

مطبوعات
مكتبة الملك فهد الوطنية
السلسلة الثانية
(٥٢)



أساسيات المحاكاة الحاسوبية

د . حسام بن محمد رمضان



الريـــاض
١٤٢٨ هـ / ٢٠٠٧ م

مطبوعات
مكتبة الملك فهد الوطنية
السلسلة الثانية
(٥٢)

أساسيات المحاكاة الحاسوبية

د. حسام بن محمد رمضان
أستاذ مشارك بقسم نظم المعلومات
كلية علوم الحاسب والمعلومات
جامعة الملك سعود

مكتبة الملك فهد الوطنية
الرياض ١٤٢٨هـ/٢٠٠٧م

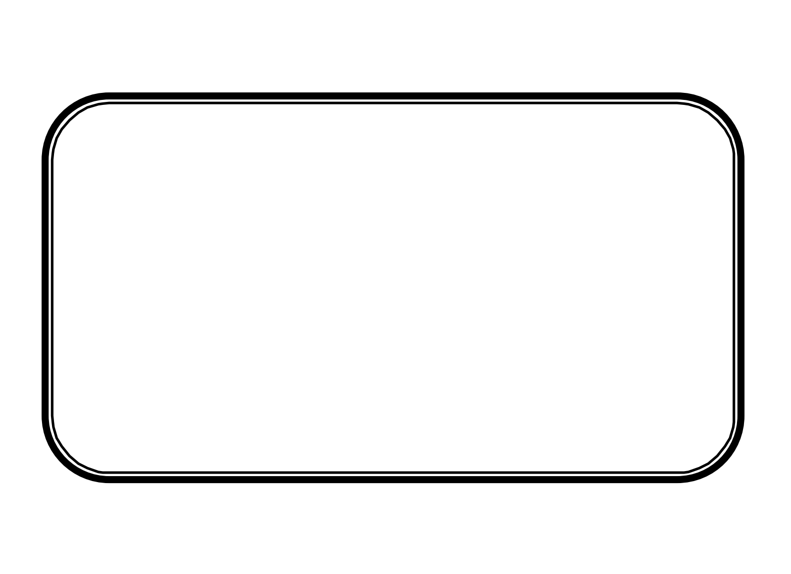
مطبوعات

مكتبة الملك فهد الوطنية

السلسلة الثانية

(٥٢)

تعنى هذه السلسلة بنشـر الدراسات والبحوث
في إطـــار علم المكتبـات والمعلومات بشكل عام



المقدمة :

يكتسب موضوع المحاكاة الحاسوبية (Computer Simulation) أهميته من اتساع قاعدة تطبيقاته في شتى مناحي الحياة ، بدءًا من المصانع إلى شبكات الاتصالات وانتهاءً بهندسة العمليات الإدارية والتدريب العسكري ، وغيرها الكثير. وتساعد نماذج المحاكاة الحاسوبية المتخصصين في هذه المجالات على أداء أعمالهم بفعالية أكبر واتخاذ قرارات أكثر إيجابية بفضل الفهم الأعمق الذي توفره هذه النماذج لواقع الأنظمة المختلفة التي تستخدم لدراستها.

وعلى الرغم من قدم هذا النوع من البرامج الذي عرف منذ أكثر من ثلاثة عقود؛ إلا أنه مازال محل اهتمام صانعي البرمجيات والذين يستثمرون اليوم الشيء الكثير في تصميم وتطوير حزم برمجيات أفضل لمحاكاة أنواع متعددة من الأنظمة. كما أنه ما زال محل اهتمام الباحثين في الجامعات ومراكز بحوث البرمجيات لاستكشاف حلول مبتكرة لمشكلات مهمة تتعلق بأساليب بناء النماذج وسرعة أدائها وكيفية الربط وإحداث التكامل والتوافق بين المحاكاة المختلفة وجعلها تعمل بفعالية بشكل جماعي، وغير ذلك من المشكلات التي سوف يؤدي التغلب عليها إلى الانطلاق إلى آفاق جديدة في تطبيق تقنيات المحاكاة والواقع الافتراضي في حياتنا اليومية بما يعود بالنفع والفائدة على البشر.

ويمثل هذا الكتاب مرجعاً علمياً عربياً لأساسيات المحاكاة الحاسوبية بما في ذلك المراحل التي يجب أن يمر بها كل من يريد تطبيق تقنية المحاكاة

على أحد الأنظمة في مجال عمله أياً كان . فيستعرض الكتاب في البداية منهجية ومراحل دراسات المحاكاة ثم يأتي عليها تباعاً بالشرح والتفصيل. ثم يتعمق هذا الشرح في أهداف كل مرحلة من مراحل دراسات المحاكاة مع تفصيل لأشهر وأهم الأساليب والمنهجيات المتبعة في تحقيق أهداف وأداء مهام كل مرحلة، وبالإضافة لذلك ، فقد ضمنت الكتاب أيضاً باباً عن بعض أبرز وأحدث الحزم البرمجية المستخدمة اليوم في ميكنة وتبسيط عملية تطوير واستخدام نماذج المحاكاة . كما احتوى الكتاب أيضا في ملحقيه على عرض مختصر لبعض الأساسيات المطلوبة والمفيدة لفهم الأساليب والمنهجيات المعروضة في الكتاب لأداء مهام مراحل دراسات المحاكاة ، وذلك في علم الإحصاء والاحتمالات (ملحق أ) وأساليب تصميم التجارب (ملحق ب) . كما تضمن الكتاب أيضاً باباً عن موضوعات متقدمة في مجال المحاكاة الحاسوبية وبالتحديد موضوع المحاكاة المتوازية والموزعة.

وأخيراً ، أسأل الله تعالى أن ينفع بهذا العمل وأن يكون عوناً لإخواني الباحثين والمهتمين بموضوع المحاكاة الحاسوبية من شتى التخصصات وأن يستفيد منه عموم الطلاب والدارسين لهذا الموضوع.

د. حسام بن محمد رمضان

الباب الأول

النماذج والمحاكاة

١-١ المقدمة :

ازداد الاهتمام بنماذج المحاكاة بشكل كبير في الآونة الأخيرة بين المهتمين بدراسات الأنظمة وبين متخذي القرار. ويعود ذلك إلى الحاجة إلى أدوات برمجية تساعد في تحقيق فهم أعمق لعمل الأنظمة وتحليل القرارات المتاحة والمرتبطة بها قبل الشروع في اتخاذ قرارات معينة ينبني عليها تخصيص للموارد. ومن هذا المنطلق، برز دور المحاكاة الحاسوبية كتطبيقات حاسوبية تسد الحاجة في هذا المجال وتساعد على نقل عالم الأنظمة الواقعي إلى عالم افتراضي يمارس فيه متخذو القرار التحليل واختبار الفرضيات على شاشات الحاسب ذات القدرة الرسومية العالية الدقة إلى أن يصلوا إلى ما يطمحون إليه من فهم وقناعات قبل اتخاذ القرارات بشأن ما يدرسونه من أنظمة.

وفي هذا الباب، سنقوم بتعريف مفهوم نماذج المحاكاة وسنعرض عدداً من تصنيفاتها المبنية على خصائصها الفنية. كما سنقوم أيضاً باستعراض الطيف الواسع من التطبيقات العملية لنماذج المحاكاة الحاسوبية، وسنقوم كذلك بعرض المراحل المختلفة لدراسات المحاكاة، كما سنتناول بالتحليل مزايا وعيوب أسلوب المحاكاة الحاسوبية.

١-٢ نماذج المحاكاة :

تعتبر النماذج (Models) من أهم الوسائل التي يستعين بها الدارسون على فهم الأنظمة المعقدة التي يصعب على المحلل استيعاب تفاصيلها بمجرد مراقبتها. ففي مثل هذه الحالات يقوم المحلل ببناء نموذج لما يريد دراسته

يكون تمثيلاً صادقاً للواقع الموجود في النظام وتجريداً لما فيه من مكونات وتفاصيل، ثم يقوم بعدها بالتعامل مع النموذج بدلاً من النظام. وتنقسم النماذج إلى عدة أنواع معروفة منها: النماذج المادية مثل المجسمات، والنماذج الرياضية مثل المعادلات والخوارزميات، والنماذج المنطقية مثل نماذج المحاكاة الحاسوبية. وتستخدم نماذج المحاكاة الحاسوبية بشكل واسع باعتبارها وسيلة لدراسة وفهم الأنظمة المعقدة والكبيرة الحجم في شتى التخصصات من أجل حل مشكلات معينة بها أو من أجل اختبار مدى تأثير فرضيات جديدة وإضافات معينة نريد أن ندرسها وندرس تأثيرها على النموذج قبل الشروع في تطبيقها فعلياً على أرض الواقع.

وتتسع تطبيقات المحاكاة الحاسوبية للكثير من التخصصات والاهتمامات منها: دعم اتخاذ القرار الإداري – التطبيقات الهندسية (شبكات الحاسب – الهندسة الصناعية – الهندسة الكيميائية) – التطبيقات العلمية (الظواهر الطبيعية – التفاعلات الكيميائية - الأنظمة البيئية والجيولوجية)، هذا بالإضافة لاستخدامات المحاكاة الحاسوبية في التعليم والتدريب.

ونموذج المحاكاة الحاسوبي هو عبارة عن تمثيل لمكونات النظام الثابتة والعلاقات التي تربط بعضها ببعض، بالإضافة إلى تمثيل منطقي لسلوكات وخصائص النظام الديناميكية على مدى فترة مراقبة زمنية محددة، وتحت فرضيات معينة تتعلق بعمل النظام ومكوناته. ويمكن نموذج المحاكاة الدارس من إجراء تجارب فرضية على النموذج بدلاً من إجرائها على النظام الحقيقي من أجل اختبار نظريات معينة حول هذا النظام أو إجابة

تساؤلات حول رد فعل النظام نتيجة لتحقق شروط أو حدوث أحداث معينة في هذا النظام. وعادة يكون استخدام نماذج المحاكاة الحاسوبية هو الاختيار الأخير بعد استنفاذ الخيارات الأخرى الممكنة من أنواع النماذج المختلفة وذلك لصعوبة تطبيق تلك النماذج، وهو ما قد يحدث عادة في حالة النماذج الرياضية للأنظمة بالغة التعقيد. وفي تلك الحالات، تأتي المحاكاة الحاسوبية كبديل قوي ومتاح لتمثيل تلك الأنظمة المعقدة وكيفية عملها على الحاسب الآلي، حيث تتيح إجراء مجموعة تجارب مصممة جيداً للإجابة على التساؤلات المطروحة حول تلك الأنظمة والتي قد تتعلق بتفاعلات النظام المدروس مع بيئته أو ربما بمدى تأثر مخرجات النظام بتغيير عدد معين من المدخلات.

وتستخدم المحاكاة الحاسوبية للأسباب التالية :

١- إجراء دراسات للإجابة عن أسئلة من نوع "ماذا – إذا" على النظام الحقيقي ، نظراً لكون نموذج المحاكاة نموذج وصفي تجريبي .

٢- يقوم نموذج المحاكاة الحاسوبي بربط العلاقات والكيانات الموجودة في النظام في صورة متكاملة تساعد متخذ القرار أو المحلل على الفهم الجيد للنظام الحقيقي .

٣- يركز نموذج المحاكاة الحاسوبي على جوانب بعينها من خصائص النظام المطلوب دراستها أو فهمها ، فلذا لا يتطلب بناءه أو استخدامه إلى معرفة شمولية بالنظام .

٤- سهولة تغيير البدائل التصميمية أو التشغيلية للنظام لاختبارها وتحديد الأنسب منها وذلك من خلال واجهات رسومية لا تتطلب معرفة متخصصة في البرمجة.

٥- إمكانية إدخال تعقيدات وتفاصيل واقعية عن المشكلات المطلوب دراستها إلى نظام المحاكاة الحاسوبية مما يزيد من دقة الحلول التي يتم التوصل إليها.

٦- القدرة على ضغط زمن النظام المحاكي بشكل كبير لدراسة سلوكات هذا النظام على مدى فترات زمنية متفاوتة الطول خلال دقائق معدودة هي فترة تشغيل برنامج المحاكاة.

٧- القدرة على دراسة الأنظمة المستقبلة التي مازالت في طور التصميم لتحديد أفضل البدائل لتصميمها وتشغيلها.

٨- سهولة تغيير فرضيات النظام وهيكله التنظيمي وبيئة عمله مع الوقت كلما استحدثت تغييرات في أي من هذه المعطيات حتى يبقى النموذج تمثيلاً جيداً للواقع.

٩- إمكانية استخدام برامج المحاكاة الحاسوبية ذات الواجهات الرسومية المتحركة وذات المؤثرات متعددة الوسائط كأدوات تعليمية وتدريبية بكفاية عالية.

وعلى الرغم من كفاية المحاكاة الحاسوبية ونجاحها باعتبارها أداة تحليلية قوية إلا أنها لا تخلو من السلبيات، وأهمها صعوبة تطوير واختبار تطبيقات المحاكاة الحاسوبية عموماً، وارتفاع تكلفة تطويرها من الناحية

الزمنية والمادية ، هذا بالإضافة إلى كونها وسيلة تقديرية تعطي حلولاً تقريبية للمشكلات المطلوب دراستها وليست أداة دقيقة كالنماذج الرياضية التي لها القدرة (غالباً) على إيجاد الحلول المثلى للمشكلات (Hill, 1996).

وتنقسم أنواع النماذج من حيث كيفية تمثيلها للنظم الفعلية إلى الأنواع التالية (Pidd,2000):

١- النماذج المادية:

هي النماذج التي تمثل الخصائص الجسمية للنظام من حيث الحجم والشكل الظاهري، ويرتبط فيها النموذج بالنظام الحقيقي بعلاقة تناسبية واضحة في هذه النواحي .

٢- النماذج الرياضية:

وتمثل العوامل المهمة في النظام على شكل مجموعة من المعادلات التي يمكن حلها آنياً لإيجاد حلول مثلى. ومن أمثلة هذه النماذج نماذج البرمجة الخطية الشائعة الاستخدام في علوم الإدارة.

٣- النماذج المنطقية:

وتستخدم خرائط التدفق (Flow Charts) لتوصيف المنطق الذي يعمل به النظام . وتستخدم تلك الخرائط لاحقاً في تصميم برامج المحاكاة الحاسوبية عن طريق ترجمتها إلى عبارات وأوامر بإحدى لغات البرمجة.

ومن الممكن أيضاً تصنيف النماذج بحسب علاقتها بالزمن إلى:

١- نماذج ساكنة (Static):

هي النماذج التي تصف خصائص النظام التي لا تتغير مع الزمن .

٢- نماذج حركية (Dynamic):

هي النماذج التي تشمل في وصفها للنظام الجوانب التي تتغير بمرور الزمن .

كما يمكن أيضاً تصنيف النماذج بحسب عشوائيتها إلى:

١- نماذج محددة (Deterministic):

وهي النماذج التي تصف الخصائص غير العشوائية في النظام .

٢- نماذج احتمالية (Stochastic):

وهي النماذج التي تصف الخصائص والظواهر العشوائية في النظام وتستخدم لذلك نظرية الاحتمالات مفاهيم العمليات والمتغيرات العشوائية.

وأخيراً يمكن تصنيف النماذج أيضاً بحسب اتصالها وتقطعها كالتالي:

١- نماذج متصلة (Continuous):

وهي النماذج التي تصف حالة النظام من خلال مجموعة من متغيرات الحالة التي تتغير مع الزمن كدالة متصلة في محور الزمن .

٢- نماذج متقطعة (Discrete):

وهي النماذج التي تصف حالة النظام من خلال مجموعة من متغيرات الحالة التي تتغير مع الزمن كدالة متقطعة مع محور الزمن.

ونلاحظ في التصنيف الأخير أن متغيرات الحالة ، سواء متصلة أو متقطعة مع الزمن ، قد تكون بحد ذاتها ذات قيم متصلة أو متقطعة ، كما يوضح ذلك الشكل رقم (١-١).

متغير الحالة

متغير الحالة

مجموعة
القيم
المتاحة
لمتغير
الحالة

نطاق
القيم
المتاحة
لمتغير
الحالة

الزمن

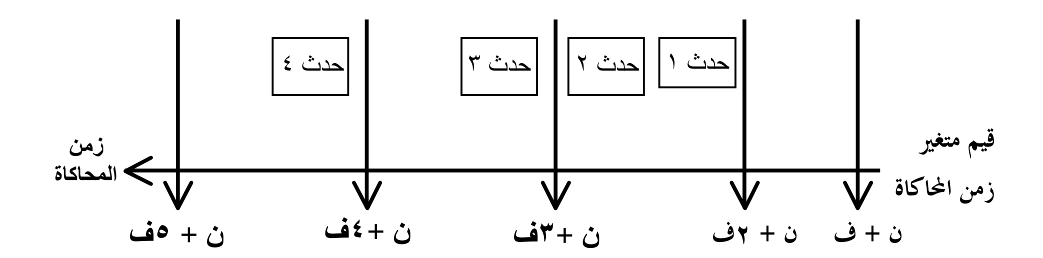
الشكل رقم (١-١) : تصنيف النماذج بحسب اتصالها أو تقطعها مع الزمن

كما تتفاوت النماذج أيضاً من حيث أسلوب معالجتها للزمن وكيفية تمثيل تغيره في المحاكاة. وهناك لمعالجة الزمن في نماذج المحاكاة الحاسوبية أسلوبان معروفان: أسلوب "تقطيع الزمن" وأسلوب "القفز على الأحداث" (Pidd, 2000) . أما أسلوب تقطيع الزمن فيتم فيه تحريك متغير الزمن في المحاكاة للأمام بفترات متساوية ، مع تحديث قيم متغيرات الحالة في النموذج حسب المستجدات والأحداث المجدولة خلال هذه الفترة. ويعيب هذا الأسلوب صعوبة تحديد القيمة المثلى لفترة تحريك الزمن ، حيث يؤثر عدم الدقة في تحديدها على دقة نتائج المحاكاة - فيما لو كانت أكبر من اللازم – أو قد يؤدي إلى بطء تنفيذ برنامج المحاكاة – فيما لو كانت أصغر من اللازم.

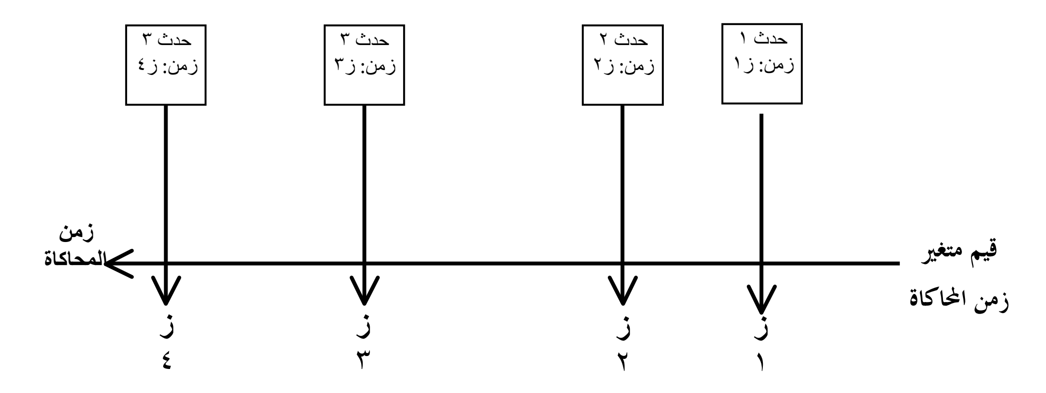
وأما أسلوب القفز على الأحداث – وهو الأكثر شيوعاً – فهو يعتمد على تحريك زمن المحاكاة للأمام بفترات غير متساوية عن طريق معالجة الأحداث المجدولة في برنامج المحاكاة واحدة تلو الأخرى، ومن ثم تحريك زمن المحاكاة بناء على الأزمنة المختلفة لكل حدث مجدول يتم معالجته، وليس عن طريق تحديد فترات زمنية متساوية ومعالجة ما بداخلها من أحداث كما في أسلوب تقطيع الزمن. ويتميز هذا الأسلوب بكفايته من حيث تركيزه لنشاط برنامج المحاكاة على الأعمال المثمرة (معالجة الأحداث

المجدولة) على العكس من الأسلوب الآخر الذي يوجه جهد برنامج المحاكاة بنسبة كبيرة إلى اختبار الفترات الزمنية المتتالية بشكل روتيني رتيب لمعرفة ما إذا كانت تحوي أحداثاً أم لا. ويوضح الشكل رقم (١-٢) تمثيلاً لكيفية تحريك الزمن في كل من الأسلوبين السابقين.

(أ): أسلوب تقطيع الزمن بفترة ف



(ب): أسلوب القفز على الأحداث



الشكل رقم (١-٢) تمثيل كيفية أسلوبي تحريك الزمن في المحاكاة

١-٢-١ المحاكاة الحاسوبية الشيئية:

بدأت فكرة المحاكاة الشيئية تاريخياً بظهور لغة المحاكاة (SIMULA) في الستينيات الميلادية ، وكانت هي اللبنة الرئيسة لظهور أفكار البرمجة الشيئية فيما بعد. وفي المحاكاة الشيئية ينظر إلى النظام باعتباره مجموعة من الكائنات التي تتفاعل فيما بينها داخل حدود النظام، ولذا فهي تتفق وتتجانس مع رؤية المحاكاة المتقطعة الأحداث (DES) للأنظمة، وهو ما جعل مصممو نماذج المحاكاة - وخصوصاً بأسلوب تفاعل العمليات (Process Interaction) - يوظفون المفاهيم الشيئية بسلاسة في بناء تصاميم أفضل لنماذجهم، ويستفيدون بذلك من لغات البرمجة الشيئية الحديثة في كتابة برامج المحاكاة التي تفعل هذه النماذج . وفي المحاكاة الشيئية يتم تمثيل كل فئة من كائنات نموذج المحاكاة بواسطة صنف (Class) لتمثل الأشياء التي تنتمي إلى هذا الصنف . ويحتوي تعريف الصنف على عدد من الحقول لتسجيل قيم المتغيرات التي تعبر عن حالة الشيء بالإضافة إلى عدد من العمليات والتي تسمى أيضاً طرقاً (Methods) يمثل كل منها عملية من مجموعة العمليات التي يقوم بها الأشياء التي تنتمي إلى الصنف، وتمثل في مجملها السلوكات العامة لهذه الأشياء .

وكما هو الحال في الأنظمة البرمجية الشيئية ، فإن المحاكاة الشيئية تستفيد من مفهوم التوريث (Inheritance) الذي يسمح للصنف الواحد بأن يعرف على أنه حالة خاصة من واحد أو أكثر من الأصناف العامة "الوالدة"

مما يقلل في النهاية من الجهد البرمجي في كتابة برنامج المحاكاة من خلال إعادة استخدام تعريفات هذه الأصناف في تعريف الصنف الوليد.

١-٣ مراحل وخطوات دراسة المحاكاة الحاسوبية:

فيما يلي نستعرض المراحل التي تمر بها دراسة المحاكاة الحاسوبية والخطوات الواجب اتباعها لإنجازها (Hoover and Perry, 1990) . ويوضح الشكل رقم (١-٣) تصويراً لهذه المراحل والعلاقة بينها.

١- تعريف المشكلة:

من المهم قبل الشروع في تنفيذ دراسة المحاكاة الحاسوبية أن يسبقها مرحلة تخطيطية يتم فيها تحديد أهداف الدراسة والجدوى منها ومدى ملاءمة أسلوب المحاكاة كوسيلة لتحقيق المطلوب. وكذلك، يجب أن يتم في هذه الخطوة تعريف المتغيرات المحددة للقرار (decision variables) والمؤثرات (parameters) التي لا يمكن التحكم بها، ومؤشرات الأداء (performance indicators) التي سيتم قياس الأداء بمراقبة قيمها.

٢- بناء النموذج المبدئي:

وتشمل هذه الخطوة تحديد المكونات الثابتة للنموذج من العناصر التالية (Pidd, 1998) :

أ- **كيانات** (entities) وتعريفها على أنها أي شيء ذي علاقة في النظام له صفات (attributes) ويقوم بأداء أنشطة (activities).

ب- **أحداث** (events) وهي الوقائع الزمنية التي ينشأ عنها تغيير حالة النظام.

ج- متغيرات الحالة (state variables) وهي مجموعة المتغيرات اللازمة لتوصيف حالة النظام في أي وقت حسب أهداف الدراسة.

٣- جمع البيانات وتحليلها:

ويتم في هذه الخطوة جمع البيانات عن النظام إما يدوياً أو آلياً حسب أهداف الدراسة ودرجة الدقة المطلوبة في مقابل التكلفة المادية والزمنية. كما يتم أيضاً تحليل تلك البيانات إحصائياً من أجل استنتاج النمط الإحصائي للبيانات باستخدام إحدى التوزيعات الإحصائية المعروفة. وفي حالة تعذر ذلك، من الممكن استخدام عينة البيانات المجموعة مباشرة في برنامج المحاكاة، ولو أن ذلك يجعل تنفيذ البرنامج بطيئاً.

٤- تحويل النموذج إلى برنامج حاسوبي:

تتم في هذه الخطوة عملية تطوير لبرنامج المحاكاة الحاسوبي انطلاقاً من النموذج المبدئي والمعلومات الإحصائية التي تم استخلاصها عن النظام. وتشمل عملية التطوير تصميم البرنامج كمخطط ثم اختيار لغة البرمجة وأدوات التطوير ثم البرمجة والتطوير.

وبالتحديد، تتم عملية تحويل للنموذج النظري المطلوب محاكاته، والذي تم إعداده بناء على مسح وتحليل لهذا النظام، إلى نموذج حاسوبي يمثل تصميماً مفصلاً لبرنامج المحاكاة المطلوب تطويره. وتتطلب هذه الخطوة بالضرورة تثبيت نوعية نموذج المحاكاة واختيار أسلوب بنائه من بين الأساليب المعروفة لدى المتخصصين. وتتراوح هذه الأساليب بين النمذجة باستخدام النماذج الإعلانية (Declarative Models) أو النماذج الوظيفية (Functional Models) – على سبيل المثال – أو غيرها من أنواع نماذج المحاكاة المختلفة (Fishwick, 1995). وبتثبيت نوعية وأسلوب بناء نموذج

المحاكاة نكون قد ثبتنا أيضاً الكثير من التفاصيل المتعلقة بالكيفية التي يتم بها توصيف الخصائص الديناميكية للنظام المطلوب محاكاته، وبالكيفية التي يتم بها تحريك العامل الزمني في المحاكاة الحاسوبية، فضلاً عن الهيكل العام لبرنامج المحاكاة الذي سيُطور بناء على نموذج المحاكاة الحاسوبي. وبعد تثبيت نوعية نموذج المحاكاة ينبغي أيضاً أختيار أسلوب من أساليب هيكلة برنامج المحاكاة والتي عادة ما تتراوح بين أسلوب قوائم الأحداث (Event List) أو أسلوب تفاعل العمليات (Process Interaction) . بعد ذلك، على المصمم تحديد نوعية برنامج المحاكاة. ونوعية برنامج المحاكاة هنا تمليها نوعية بيئة المحاكاة المستخدمة لتطوير البرنامج. ويقصد ببيئة المحاكاة البرنامج الحاسوبي المتكامل الذي يوفر للمبرمج واجهة رسومية وتسهيلات برمجية عدة تمكنه من تطوير برنامج المحاكاة المطلوب بسهولة ويسر. وتوجد اليوم نوعيات مختلفة من بيئات المحاكاة الحاسوبية ذات نوعيات تتراوح بين البرمجة اليدوية بإحدى لغات برمجة المحاكاة إلى البرمجة المرئية (Visual Programming) باستخدام القوالب (Blocks) التي ترمز إلى مكونات نموذج المحاكاة. واختيار أسلوب بنائه من بين الأساليب المعروفة لدى المتخصصين. وتتراوح هذه الأساليب بين النمذجة باستخدام النماذج الإعلانية (Declarative Models) أو النماذج الوظيفية (Functional Models) – على سبيل المثال – أو غيرها من أنواع نماذج المحاكاة الحاسوبية المختلفة (Fishwick, 1995). وبتثبيت نوعية وأسلوب بناء نموذج المحاكاة الحاسوبية نكون قد ثبتنا أيضاً الكثير من التفاصيل المتعلقة بالكيفية التي يتم بها توصيف الخصائص الديناميكية للنظام المطلوب محاكاته، وبالكيفية التي يتم بها تحريك العامل الزمني في المحاكاة، فضلاً عن الهيكل

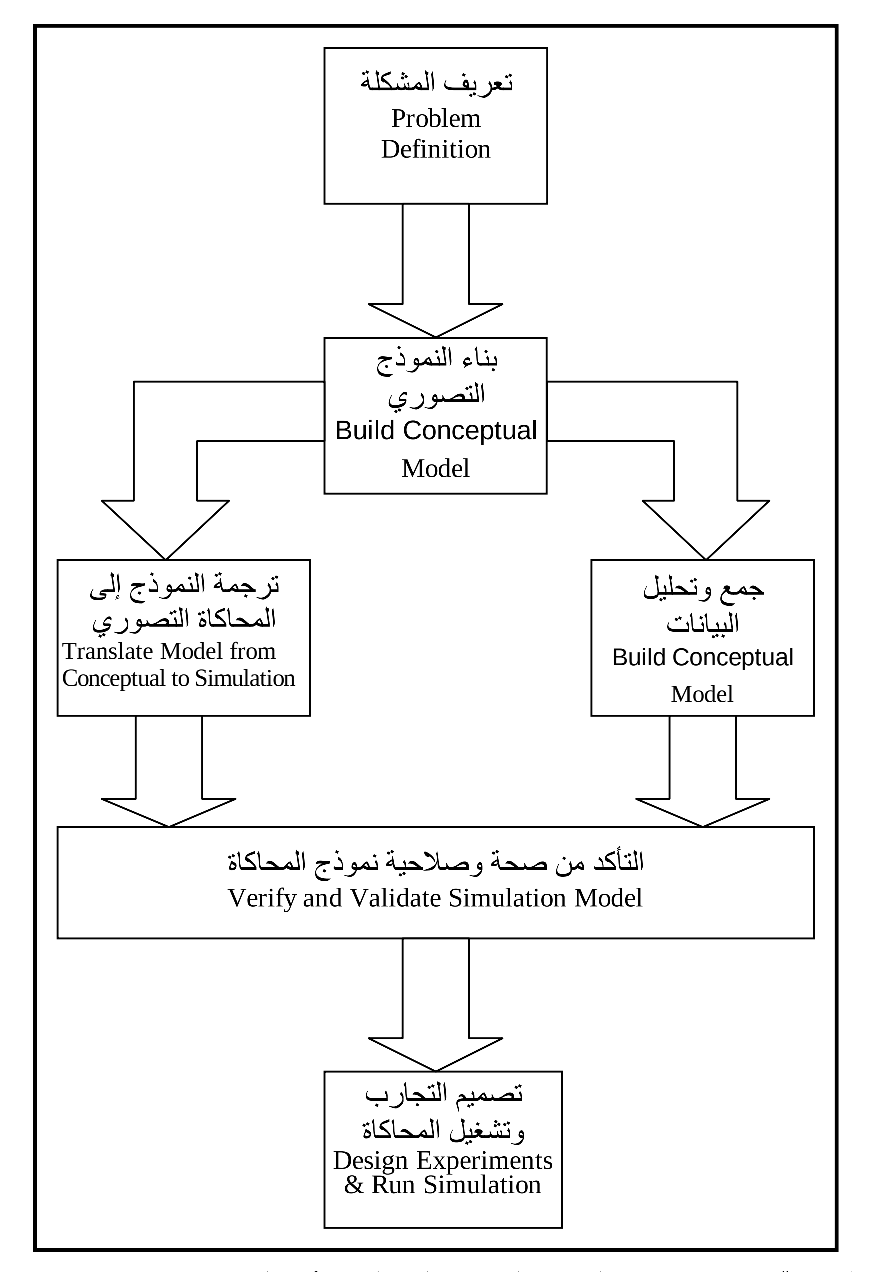
العام لبرنامج المحاكاة والذي سيُطور بناء على نموذج المحاكاة الحاسوبي. وبعد تثبيت نوعية نموذج المحاكاة ينبغي أيضاً اختيار أسلوب من أساليب هيكلة برنامج المحاكاة والتي عادة ما تتراوح بين أسلوب قوائم الأحداث (Event List) أو أسلوب تفاعل العمليات (Process Interaction). بعد ذلك، على المصمم تحديد نوعية برنامج المحاكاة. ونوعية برنامج المحاكاة هنا تمليها نوعية بيئة المحاكاة الحاسوبية المستخدمة لتطوير البرنامج.

ويقصد ببيئة المحاكاة البرنامج الحاسوبي المتكامل الذي يوفر للمبرمج واجهة رسومية وتسهيلات برمجية عدة تمكنه من تطوير برنامج المحاكاة المطلوب بسهولة ويسر. وتوجد اليوم نوعيات مختلفة من بيئات المحاكاة الحاسوبية ذات نوعيات تتراوح بين البرمجة اليدوية بإحدى لغات برمجة المحاكاة إلى البرمجة المرئية (Visual Programming) باستخدام القوالب (Blocks) التي ترمز إلى مكونات نموذج المحاكاة.

٥- التحقق من صحة النموذج وصلاحيته:

تهدف عملية التحقق من صحة النموذج (Model Verification) إلى التأكد من الترابط الداخلي للنموذج الحاسوبي وتوافقه مع النموذج المبدئي. أما عملية التأكد من الصلاحية (Model Validation) فتهدف إلى مضاهاة النموذج الحاسوبي بالنظام الأصلي للتأكد من جودة تمثيل نموذج المحاكاة الحاسوبي للنظام الأصلي، سواء من الناحية الظاهرية أو من حيث تماثل عملية التحويل (Transformation) التي تجري بداخله على المدخلات لتحويلها إلى مخرجات. وتعتبر عملية الاختبار والتأكد من الجودة من المهام الرئيسة في أي دراسة محاكاة بعد انتهاء عملية البرمجة والتطوير لبرنامج المحاكاة الذي يفعل نموذج محاكاة النظام. وتشمل هذه العملية اختبار

البرنامج المطور من حيث صحة البرمجة منطقياً ووظيفياً. كما تشمل أيضاً التأكد من صلاحية برنامج المحاكاة باعتباره نموذجًا يعبر تعبيراً دقيقاً عن سلوكات النظام الأصلي. ولإنجاز عملية الاختبارات، يلزم القيام بالخطوات الرئيسة التالية (Pooch, 1993) :



الشكل رقم (١-٣): خطوات ومراحل دراسة المحاكاة الحاسوبية

أ- التأكد من صحة البرنامج منطقياً ووظيفياً :

تهدف هذه الخطوة إلى التأكد من أن برنامج المحاكاة صحيح من الناحية البرمجية. وتنقسم هذه الخطوة إلى عمليتين: عملية التأكد من صحة البرنامج منطقياً وعملية التأكد من صحة البرنامج وظيفياً(Law and Kelton, 2000) .

أ-١ عملية التأكد من صحة البرنامج منطقياً :

تعتمد خطوة التأكد من صحة البرنامج منطقياً على مراجعة التصميم الذي طور البرنامج بناء عليه، بما في ذلك مراجعة مخططات التدفق (Flow Charts) والتي تحدد التفاصيل البرمجية التي ينبني عليها البرنامج. وتتم هذه العملية فور الانتهاء من تطوير البرنامج بصورة تكرارية إلى أن يتم التوصل إلى قناعة بخلو البرنامج من الأخطاء المنطقية والبرمجية.

أ-٢ عملية التأكد من صحة البرنامج وظيفياً :

تعتمد خطوة التأكد من صحة البرنامج وظيفياً على تشغيل البرنامج بصورة تجريبية لمرات عدة، وفي كل مرة يتم التركيز على أحد المخرجات المعطاة في نهاية البرنامج أو أحد الوظائف الأساسية المتوقع من البرنامج إنجازها أثناء التشغيل. وتهدف هذه العملية إلى التأكد من استيفاء البرنامج للوظائف والأهداف التي وضعت له والتأكد من عدم إغفال أي منها. ويتم ذلك بعد الانتهاء من تطوير البرنامج للتأكد من أن جميع المخرجات المطلوبة والإحصاءات المطلوب جمعها أثناء تشغيل البرنامج قد تضمنها البرنامج الحالي ويقوم بتدوينها بنجاح.

ب- التأكد من صحة النموذج :

يتم في هذه الخطوة التأكد من توافق البرنامج الذي تم تطويره وتم التأكد من صحته برمجياً ووظيفياً في الخطوة السابقة مع تفاصيل النموذج النظري والذي تم وضعه بناء على نتائج عملية تحليل النظام. وبالتحديد، فإن هذه الخطوة تضاهي بين النموذج الذي يفترضه البرنامج في ثناياه وبين النموذج النظري الذي يفترضه المحلل من واقع تحليله للنظام الأصلي. وعادة يتم اتباع الأساليب الآتية لإنجاز هذه المهمة:

ب-١ أسلوب التتبع الهيكلي: ويعتمد هذا الأسلوب على إعداد ملخص للفرضيات التي تم وضعها عن النظام وطريقة عمله من واقع الدراسة التحليلية التي تم إجراؤها عن هذا النظام وخصائصه، ومن واقع البيانات الإحصائية التي تم جمعها عن عمل هذا النظام في السابق. ويحتوي هذا الملخص أيضاً على توصيف لأهم عناصر ومكونات النظام والعلاقات التي تربط بعضها ببعض، كما يحتوي على ذكر أي فرضيات تقريبية وضعت لتبسيط النموذج، بالإضافة إلى أية معلومات عن الخصائص الإحصائية للمدخلات التي تم تعريفها في النموذج. بعد ذلك، يتم عمل تتبع منظم وهيكلي لنموذج المحاكاة الحاسوبي المطور خطوة بخطوة ومكون بمكون مع الرجوع إلى ملخص الفرضيات للتأكد من مطابقة تصميم النموذج لتلك الفرضيات.

ب-٢ الأسلوب الكمي: وفي هذا الأسلوب يتم مراجعة مخرجات البرنامج المرحلية (قيم المتغيرات الداخلية في البرنامج) من أجل التأكد من معقولية هذه النتائج.

ب-٣ الأسلوب الكيفي: ويعتمد هذا الأسلوب على الاستعانة بخاصية تحريك النموذج أثناء المحاكاة (Model Animation) والتي تتيح مشاهدة رسوم متحركة على الشاشة ترمز لكل مكون من مكونات النموذج وتعبر عما يحدث أثناء تنفيذ البرنامج من عمليات وتفاصيل ومشكلات.

أما خطوة التأكد من صلاحية النموذج (Model Validation) فتهدف إلى التأكد من التوافق بين برنامج المحاكاة وما ينطوي عليه من نموذج للنظام الحقيقي مع تفاصيل ذلك النظام الحقيقي، مما يدلل على أن برنامج المحاكاة صالح لأن يكون بديلاً عن هذا النظام الحقيقي لأغراض صنع القرار. وتعتمد هذه الخطوة على إجراء اختبارات للتأكد من مطابقة عملية تحويل المدخلات إلى مخرجات (Input-Output Transformation) في كل من البرنامج والنظام الحقيقي. والطريقة المتبعة عادة لذلك هي المقارنة بين مخرجات النظام الحقيقي (من واقع المعلومات المتوفرة عنه تحت ظروف تشغيلية ومدخلات معينة) وبين المخرجات النظيرة التي يتنبأ بها نموذج المحاكاة تحت نفس الظروف التشغيلية ونفس قيم المدخلات. فإذا ما كانت نتيجة المقارنة قريبة بدرجة كافية لأغراض الدراسة يتم إجازة النموذج، وإذا كانت النتيجة غير ذلك تتم مراجعة النموذج ومحاولة تحسينه ثم إجراء كافة الاختبارات عليه مرة أخرى.

٦- تصميم التجارب واستخراج النتائج :

وفي آخر مراحل دراسة المحاكاة الحاسوبية يتم تصميم سلسلة من التجارب تهدف إلى إيجاد أفضل مجموعة من قيم متغيرات القرار التي يتوقع نموذج المحاكاة الحاسوبي أن ينتج عن استخدامها في النظام الأصلي تحقيق قيم مقبولة من مؤشرات الأداء حسب أهداف الدراسة. ومن الممكن في حالات خاصة استخدام أساليب تكرارية للبحث عن قيم مثلى من مؤشرات الأداء؛ إلا أن هذه الحالات تحتاج إلى توفر شروط كثيرة في النظام حتى يصبح من الممكن استخدام تلك الأساليب والحصول على نتائج صحيحة منها. وتجدر الإشارة إلى أن عملية تطوير برامج المحاكاة باستخدام لغات البرمجة التقليدية تعتبر عملية شاقة ومعقدة، حيث يحتاج المبرمج إلى القيام بالكثير من المهام المطلوبة يدوياً؛ إلا أنها توفر أكبر قدر من المرونة و القدرة على النمذجة. وعلى النقيض، فإن ما يسمى بلغات المحاكاة توفر قدراً من التناظر بين مكونات الأنظمة المطلوب محاكاتها والتراكيب البرمجية، مما يوفر على المبرمج الكثير من الجهد نظراً؛ لأنه في هذه الحالة يستخدم وحدات بناء كبيرة و سابقة التجهيز، مما يعني إنتاجية وكفاية أعلى؛ إلا أن ذلك يأتي على حساب المرونة والقدرة على النمذجة وأيضاً سرعة تشغيل البرنامج. ومن الأمثلة على لغات وأدوات المحاكاة الحاسوبية الشائعة:

. SIMAN, SIMSCRIPT, Arena, Extend

الباب الثاني

تصميم نماذج المحاكاة

٢-١ المقدمة :

يعرف النموذج على أنه "تجريد للواقع بغرض توصيف نظام معين". فعند محاولة التعامل مع أنظمة معقدة أو كبيرة الحجم بغرض فهم خصائصها أو بغرض إيجاد حلول للمشكلات القائمة بها، يواجه المحلل بصعوبة الإحاطة بكل التفاصيل المتعلقة بهذه الأنظمة أو استيعابها بمجرد النظر فيما يتوفر عنها من معلومات أو بيانات. وعندئذ، ينبغي على المحلل أو الدارس للنظام أن يقوم ببناء نموذج مناسب لنوع النظام ليساعده على الوصول إلى تصور أفضل عن عمل هذا النظام ومكوناته وخصائصه.

وقد سبق في الباب الأول استعراض أنواع النماذج المختلفة التي تتراوح بين المجسمة والرياضية والحاسوبية، والتي يختار بينها المحلل أو الدارس بحسب نوع النظام المطلوب دراسته. كما سبق تلخيص الفوائد التي توفرها عملية بناء النماذج الحاسوبية والتي تشمل تبسيط الأنظمة المعقدة والكبيرة أمام الدارسين وتوحيد الرؤية حول النظام بين الدارسين المختلفين. كما تم أيضاً استعراض أنواع وتصنيفات مختلفة لنماذج المحاكاة الحاسوبية تتفاوت في مدى التجريد الذي تفرضه وتعتمد على طبيعة النظام وخصائصه وما إذا كان - على سبيل المثال - متصلاً أو منقطعًا مع الزمن أو مع قيم متغيرات الحالة. وسوف نقوم لاحقاً في هذا الباب بالدراسة التفصيلية لكل من هذه الأنواع مع استعراض أمثلة عن استخداماتها في مجال المحاكاة الحاسوبية.

٢–٢ نماذج النظم الثابتة زمنياً :

تأتي عملية تعريف المكونات الثابتة زمنياً (Static) في النظام المطلوب محاكاته كإحدى أولى مراحل عملية بناء نموذج المحاكاة. ويبدأ المحلل

العملية من واقع ما يتوافر له من معلومات وصفية عن النظام وخصائصه، والتي عادة ما تكون في إحدى صورتين: الوصف الكلامي أو الوصف الرسومي. وتشتمل عملية تعريف المكونات الثابتة زمنياً على تحليل هذه المعلومات والحصول منها على توصيف لخصائص النظام التي لا تتأثر مع الزمن، مثل تعريف الكيانات الرئيسة وصفاتها وعلاقاتها مع بعضها البعض. وعادة ما تستخدم في ذلك مفاهيم النمذجة الشيئية – والتي سبق التعريف بها في الباب الأول – باعتبارها وسيلة فعالة لتمثيل الكيانات المكونة للنموذج وصفاتها (Attributes) وطرقها (Methods) التي تحدد سلوكاتها والأعمال التي تستطيع القيام بها. ويقوم المحلل - من أجل الوصول إلى تحديد هذه المكونات – بمحاولة الربط بين المعلومات المختلفة المتوافرة حول النظام، واستيضاح أي غموض أو لبس حول تفاصيله حتى يصل تدريجياً بالنموذج من صورة وصف لفظي أو رسومي إلى صورة نموذج تصوري واضح يمكن منه الانتقال بسلاسة إلى مرحلة تطوير برنامج المحاكاة، نظراً لأن النموذج التصوري خاضع لنظرية ناضجة ومتكاملة هي نظرية النمذجة الشيئية.

٢-٢-١ كيفية تعريف النماذج التصورية الشيئية :

للانتقال من المعلومات الوصفية الكلامية عن النظام إلى تعريف نموذج شيئي يحدد مكونات النظام الثابتة زمنياً، يجب على المحلل تحليل النصوص الوصفية المتوافرة عن هذا النظام بغرض استخلاص الأصناف (Classes) الرئيسة المكونة للنظام، والتعرف إلى أحداث النظام، وغيرها من مكونات

النموذج. ويحتاج المحلل لإنجاز هذه المهمة إلى اتباع بعض القواعد البديهية لاستخلاص هذه المكونات من متن النص الوصفي، ومنها على سبيل المثال:

١- أن الأحداث تكون عادة في مواضع التفاعل بين الكيانات المختلفة في النموذج، والتي يحدث عندها تغير مهم في حالة النظام أو قيمة مخرجاته.

٢- تدل الأسماء عادة على احتمالية تعريف صنف.

٣- تدل الصفات على احتمالية تعريف صفة لصنف أو شيء من صنف.

٤- تدل الأفعال على احتمالية وجود طريقة (Method).

وجدير بالذكر أنه قد يكون وصف النظام المتوافر للمحلل ليس فقط على هيئة نص بل قد يشمل أيضاً الصور الوصفية والتي قد تتراوح بين الرسم الكروكي التقريبي إلى المخطط التفصيلي، وربما يشتمل أيضاً على رسوم تصورية لتوضيح العلاقات السببية بين الأحداث وحالات النظام (Fishwick, 1995).

٢ - ٣ نماذج النظم الديناميكية :

بصفة عامة، يتكون أي نموذج ديناميكي من عدة مكونات رئيسة:

- الحدث (Event).
- الحالة (State).
- المدخلات (Input).

- المخرجات (Output).
- الزمن (Time).

ويعرف الحدث عادة على أنه : "نقطة زمنية يحدث عندها تغير في حالة النظام"، وتعرف الحالة على أنها : "وصف للنظام على مدى فترة زمنية". ورياضياً، تعرف حالة النظام كمتجه من n بعد ، ويرمز لها بعدد n رقم مرتب كالتالي :

$$S=(s_1, s_2, \ldots, s_n)$$

حيث يمثل n عدد مكونات حالة النظام، وهو أيضاً عدد متغيرات الحالة (State Variables) ، ويمثل s_i متغير الحالة رقم i من متجه حالة النظام. وتعرف المدخلات على أنها: "كميات متغيرة ذات تأثير على حالة النظام ولا تقع داخل حدوده". وفي حالة ما إذا كان المدخل ذي قيمة ثابتة، فعندئذ يسمى مؤثراً (Parameter). أما المخرجات فهي: " كميات يمكن مراقبتها وتعبر عن سلوكات وأداء النظام"، وتعتبر من الناحية الرياضية دالة في متغيرين هما المدخلات والحالة. ويعرف الزمن على أنه: "متغير متقطع أو متصل يعبر عن مفهوم الزمن في النظام الديناميكي، والذي من شأنه أن تتغير حالته مع زيادة قيمة الزمن". وتنقسم نماذج النظم الديناميكية بحسب كيفية تحريك الزمن بها إلى نماذج متقطعة زمنياً يتم فيها زيادة الزمن بمقادير محددة بصورة دورية ، ونماذج متصلة زمنياً يتم فيها زيادة الزمن بمقادير متناهية الصغر بحيث يكون محور الزمن متصلاً من الناحية الرياضية.

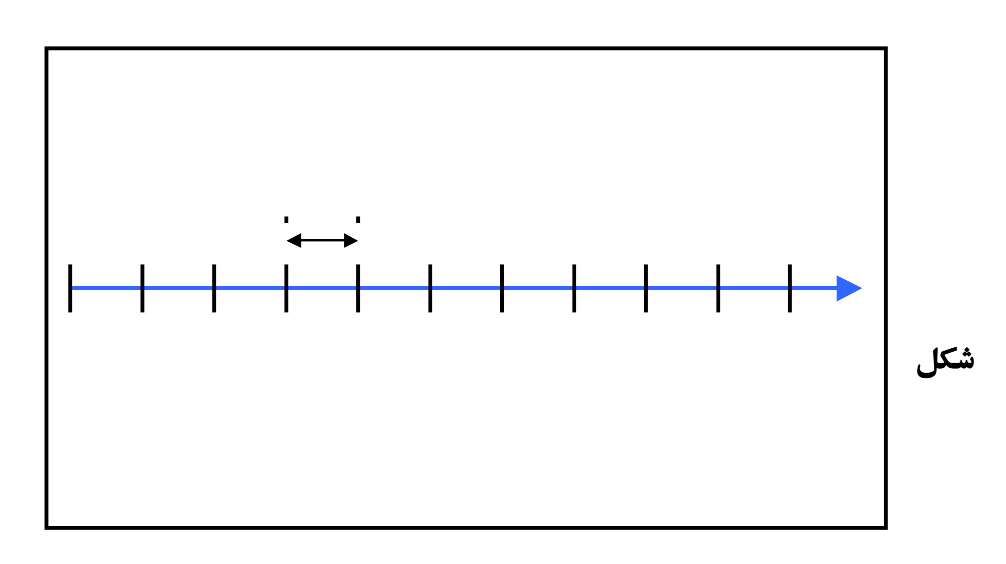
٢-٤ أساليب معالجة زمن المحاكاة :

يقصد بمعالجة زمن المحاكاة عملية التحكم في قيمة ساعة المحاكاة (Simulation Clock) والتي هي قيمة من الزمن تعبر عن "*الزمن الحالي*" في نطاق زمن المحاكاة والذي وصل إليه برنامج المحاكاة من خلال تنفيذه حتى لحظة الاستعلام عن قيمة ساعة المحاكاة. ولذا، فمن الواضح أن قيمة ساعة المحاكاة سوف تبدأ عند بداية تشغيل برنامج المحاكاة عند قيمة أولية ، ثم يقوم البرنامج بزيادة قيمتها بصورة مطردة إلى أن تبلغ القيمة العظمى المحددة من المحلل والتي عندها يجب أن تتوقف المحاكاة. وتعرف عملية زيادة قيمة ساعة المحاكاة تدريجياً من قبل البرنامج بعملية معالجة زمن المحاكاة. وهناك طريقتان لمعالجة زمن المحاكاة: طريقة تشريح الوقت (Time Slicing) وطريقة الحدث التالي (Next Event).

٢-٤-١ طريقة الحدث التالي :

وتعتمد هذه الطريقة على تقسيم فترة محاكاة النظام الكلية إلى شرائح زمنية ذات طول متساو (ط) يحدده المبرمج. ثم يقوم البرنامج بناء عليه بمسح هذه الفترات منذ بداية المحاكاة حتى نهايتها واحدة تلو الأخرى عن طريق قراءة الأحداث المجدولة سلفاً والتي تقع أزمنتها بداخل الفترة الحالية وتنفيذ إجراءات هذه الأحداث بترتيب قيمة أزمنتها الواقعة بداخل الفترة الحالية. ثم يغير البرنامج الفترة إلى الفترة التالية ويقرأ مرة أخرى الأحداث التي تقع فقط بداخل الفترة التالية وينفذ إجراءاتها بالترتيب، وهكذا. وأثناء

هذا المسح الزمني يحدّث البرنامج قيمة ساعة المحاكاة بحسب زمن آخر كل فترة حالية. أي أن قيمة ساعة المحاكاة في هذا الأسلوب سوف تتزايد بمقدار متساو هو (ط). ويوضح الشكل رقم (٢-١) فكرة تشريح الزمن.



الشكل رقم (٢-١) : أسلوب تشريح زمن المحاكاة

وبالرجوع إلى هذا الشكل نلاحظ أنه قد تكون الأحداث أحياناً موزعة زمنياً بشكل غير متكافئ أو منتظم على الشرائح الزمنية. ويؤدي المسح الزمني المتكرر للشرائح بصورة منتظمة في مثل هذه الحالة إلى إهدار وقت البرنامج في اختبار الكثير من الفترات الزمنية التي لا تحمل أحداثاً والتي يتم مسحها واختبارها على الرغم من ذلك فقط للتمشي مع متطلبات أسلوب المسح الزمني.

كما نلاحظ أن قيمة ساعة المحاكاة في هذا الأسلوب لن تعبر عن اللحظة الزمنية – في زمن المحاكاة – التي يكون عندها تنفيذ إجراء الحدث ، بل ستكون عند الحافة الكبرى للشريحة الزمنية الحالية ، مما يؤدي إلى عدم الدقة في نتائج المحاكاة . ولهذه الأسباب، فقد استبدل أسلوب تشريح الزمن بأسلوب آخر هو الأكثر شيوعاً وهو أسلوب الحدث التالي.

٢-٤-٢ طريقة الحدث التالي :

تعتمد طريقة الحدث التالي على تفادي سلبيات الأسلوب السابق عن طريق تحريك قيمة ساعة المحاكاة بحسب قيمة الزمن المجدول سلفاً للأحداث المدرجة على قائمة الأحداث المستقبلة وبالترتيب التصاعدي. ويعني ذلك أن قيمة ساعة المحاكاة سوف تكون مطابقة لقيم زمن كل حدث يتم تنفيذه؛ إلا أن هذا سوف يؤدي أيضاً إلى تغير غير سلس وغير متساو لقيمة ساعة المحاكاة ، حيث ستكون هناك قفزات في القيمة غير متساوية القيمة أو منتظمة النمط.

٢-٥ تطوير وتشغيل نماذج المحاكاة :

تعتمد نماذج المحاكاة على مفهوم "الحدث" الذي يعرف على أنه اللحظة الزمنية (في نطاق زمن المحاكاة) والتي يحدث عندها تغير ملحوظ في حالة النظام، ممثلة في تغير قيم أحد أو كل متغيرات الحالة لهذا النظام . ومتغيرات الحالة هي مجموعة المتغيرات التي تمثل قيم كميات تعبر عن حالة النظام فيما يخص أغراض دراسة المحاكاة. وتتكون نماذج المحاكاة من مجموعة من الكيانات (Entities) التي تمثل العناصر التي يتكون منها النظام وتؤدي دوراً في تحديد خصائص وسلوكات هذا النظام. وهذه الكيانات قد

تكون كيانات مؤقتة تدخل النظام وتبقى فيه فترة من زمن المحاكاة ثم تخرج منه ، أو قد تكون كيانات مستديمة تمثل أجزاء دائمة أساسية في بنية النظام. وفي أسلوب النمذجة الشيئية (Object-Oriented Modeling) تسمى الكيانات أشياءً (Object). على سبيل المثال، إذا كان النظام المطلوب محاكاته مصنعاً يتكون من آليات تقوم كل منها بمعالجة الوحدات المنتجة التي تمر عليها فتغير من حالتها حتى تصبح في صورتها النهائية، فإن الآلات المصنعة تمثل كيانات مستديمة .. بينما تمثل المنتجات الجاري تصنيعها ومعالجتها في كل آلة كيانات مؤقتة.

وعند بناء النموذج، من الممكن تصنيف الكيانات المتماثلة الخصائص (وحدات المنتجات مثلاً) من خلال مفهوم الصنف (Class) الذي يعبر عن نوع هذه الكيانات ويعرف صفاتها (Attributes) وأفعالها التي تعبر عن سلوكاتها وتفاعلاتها في داخل النظام (Methods). وتمثل المحاكاة عملية القيام بتنفيذ الإجراءات التي تعبر عن رد فعل النظام لحدوث كل نوع من أحداث هذا النظام وقت حدوث كل حدث من هذه الأحداث في زمن المحاكاة. أي أن المحاكاة تعتمد على توليد مجموعة من الأحداث المستقبلة المتوقع حدوثها في النظام بناء على النموذج التصوري الذي تم بناؤه بعد خطوة تحليل النظام . ثم تصنف هذه الأحداث بحسب نوعها ، حيث نفترض أن لكل حدث نوعًا يعبر عن معنى هذا الحدث في النظام وما يحدثه من تغيير على حالته . وكذلك ، فإن لكل حدث زمنًا (في نطاق زمن المحاكاة) يمثل الزمن الذي يتوقع أن يحدث فيه هذا الحدث. وبعدما يولد كل حدث ويولد له زمن لحدوثه في المستقبل ، علينا أن نخزن هذا الحدث مع الأحداث المستقبلة الأخرى في قائمة من الأحداث مرتبة تصاعدياً بحسب زمن الحدث

، وتسمى قائمة الأحداث المستقبلة (Future Event List) والتي تبرمج عند كتابة البرنامج كأحد تراكيب البيانات الرئيسة لبرنامج المحاكاة.

وعندما تتقدم ساعة المحاكاة – وهي الزمن الذي يمثل الوقت الحالي في نطاق زمن المحاكاة الذي وصل إليه نموذج المحاكاة أثناء عمله – إلى زمن الحدث الذي على رأس القائمة (أي الحدث ذي الزمن الأصغر) ، فإن برنامج المحاكاة يقوم باستدعاء وتنفيذ الإجراء (Routine) الذي يحتوي على مجموعة الخطوات والعمليات التي تعبر عما يحدث في النظام الحقيقي من رد فعل عند وقوع أي حدث من نوع الحدث المنفذ. وهذه الخطوات من شأنها أنها تقوم بتغيير حالة النظام وعناصره ومتغيرات حالته ، كما أنها تقوم ببعض العمليات التنظيمية التي يتطلبها برنامج المحاكاة ، مثل توليد أحداث جديدة وإضافتها بالترتيب الصحيح إلى قائمة الأحداث المستقبلة. ويسمى هذا الأسلوب في تصميم برامج المحاكاة **أسلوب قائمة الأحداث** (Event List)؛ لأنه يعتمد بشكل كبير على وجود قائمة مرتبة تصاعدياً من الأحداث المستقبلة المجدولة للحدوث في أزمنة معينة من الزمن المحاكى، والتي ينتمي كل منها إلى نوع معين من الأحداث. ويعرف لكل نوع من الأحداث إجراءً من الخطوات والعمليات المبرمجة التي تعبر عن العمليات التي تحدث فعلياً في النظام الأصلي في حال وقوع هذا النوع من الأحداث. ويتم استدعاء هذا الإجراء وتنفيذه من قبل برنامج المحاكاة كلما أتى الدور على حدوث حدث من هذا النوع – أي عندما تتقدم ساعة المحاكاة (Simulation Clock) إلى زمن حدوث هذا الحدث، والذي يكون في هذه الحالة على رأس قائمة الأحداث

المستقبلة، أي أن زمن هذا الحدث هو أقل أزمنة الأحداث المجدولة في هذه القائمة.

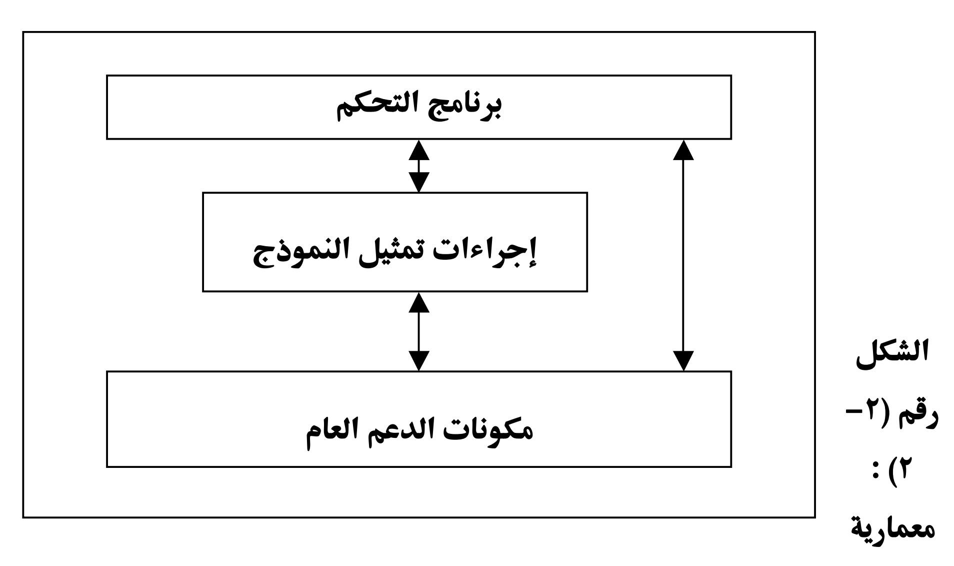
٢-٦ معمارية برامج المحاكاة :

رأينا فيما سبق أن أسلوب قائمة الأحداث لبناء برامج المحاكاة يعتمد على تمثيل رد فعل النظام عند وقوع حدث من نوع معين على شكل إجراء من مجموعة خطوات "تحاكي" رد فعل النظام عند وقوع مثل هذا الحدث. ويعمل برنامج المحاكاة من خلال تكرار العمليات التالية حتى تنتهي فترة محاكاة النظام :

- الكشف على قائمة الأحداث والتقاط أول حدث في القائمة المرتبة زمنياً بشكل تصاعدي (أي الحدث الذي يحمل أقل زمن حدوث في المستقبل)، وتحريك ساعة المحاكاة لتأخذ قيمة زمن هذا الحدث.
- استدعاء الإجراء الخاص بنوع هذا الحدث وتنفيذ هذا الإجراء، وتحديث متغيرات حالة النظام المعنية أثناء هذا التنفيذ.

ويسمى الجزء الأعلى من برنامج المحاكاة الذي يؤدي هذه المهمة ويتحكم في سير البرنامج ككل ببرنامج التحكم. ويبدو تركيب برنامج المحاكاة الكلي كما هو موضح في الشكل رقم (٢-٢)، حيث يقع برنامج التحكم في الطبقة العليا ، يليه المكون الخاص بإجراءات تمثيل النموذج، يليه المكون الخاص بالدعم العام . ويحتوي مكون إجراءات تمثيل النموذج على مجموعة الإجراءات الخاصة بكل نوع من الأحداث في النموذج والتي تعتمد

كلية على تفاصيل النموذج الذي تم بناؤه للنظام من قبل المحلل. أما مكون الدعم العام فيعبر عن مجموعة



الشكل رقم (٢-٢) : معمارية برنامج المحاكاة تحت أسلوب قائمة الأحداث

المكونات البرمجية التي تتوفر لدى برنامج المحاكاة لتوفير الوظائف العامة التي تدعم برنامج المحاكاة في عمله مثل توليد الأرقام العشوائية وجمع البيانات الإحصائية من برنامج المحاكاة أثناء التشغيل والقيام بوظائف استقبال المدخلات وعرض المخرجات والرسوميات أثناء التشغيل.

<u>مثال :</u>

ارسم خرائط منطقية لتوصيف إجراءات الأحداث المطلوبة لمحاكاة نظام صراف بنكي مكون من موظف واحد يقوم بأعمال الصراف وصف انتظار واحد لعملاء البنك للوقوف لانتظار الخدمة . افترض أن معدل وصول العملاء إلى البنك هو (ص) عميل في الدقيقة ومتوسط زمن خدمة العميل الواحد هو (خ) دقيقة .

الحل :

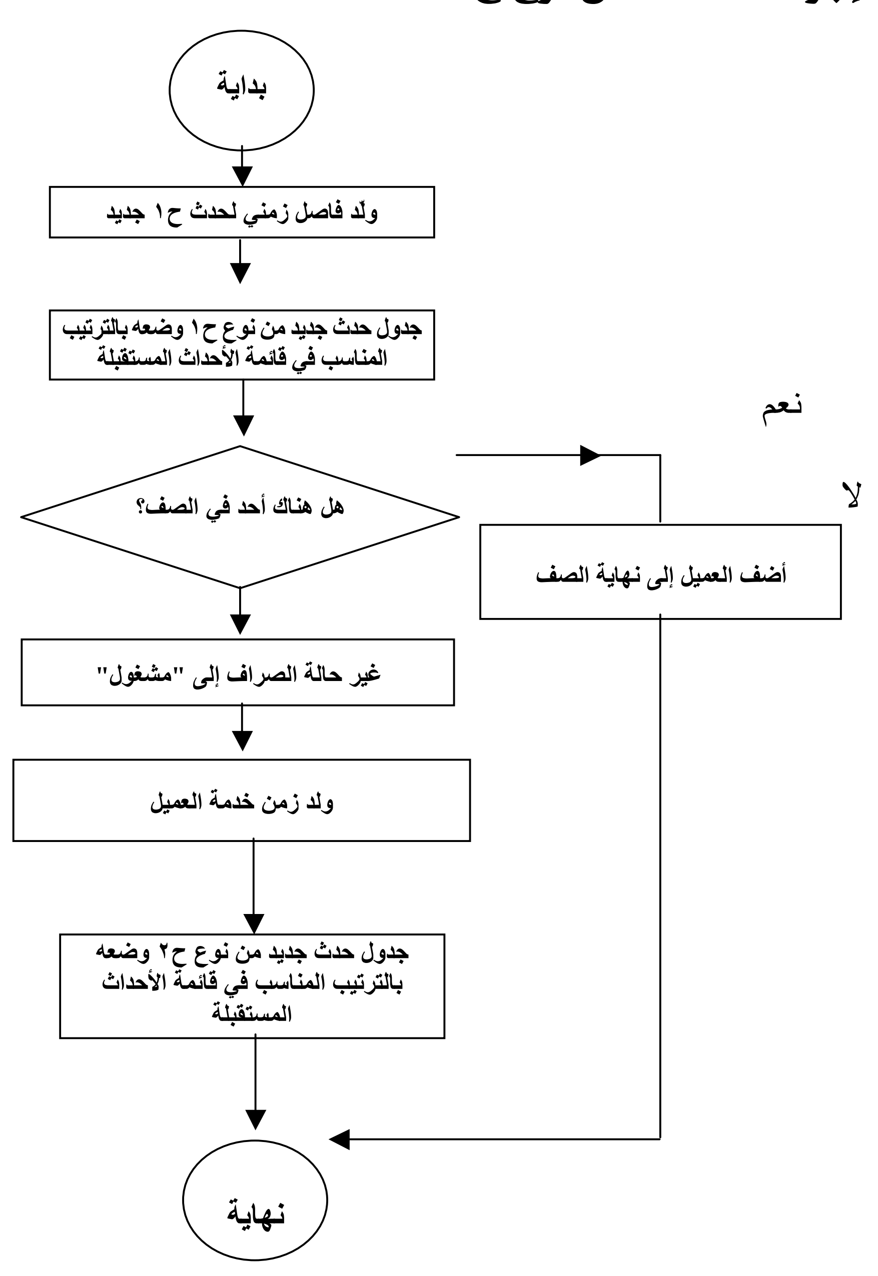
في هذا النظام ، يمكننا التعرف إلى النوعين التاليين من الأحداث:

ح١ : وصول عميل .

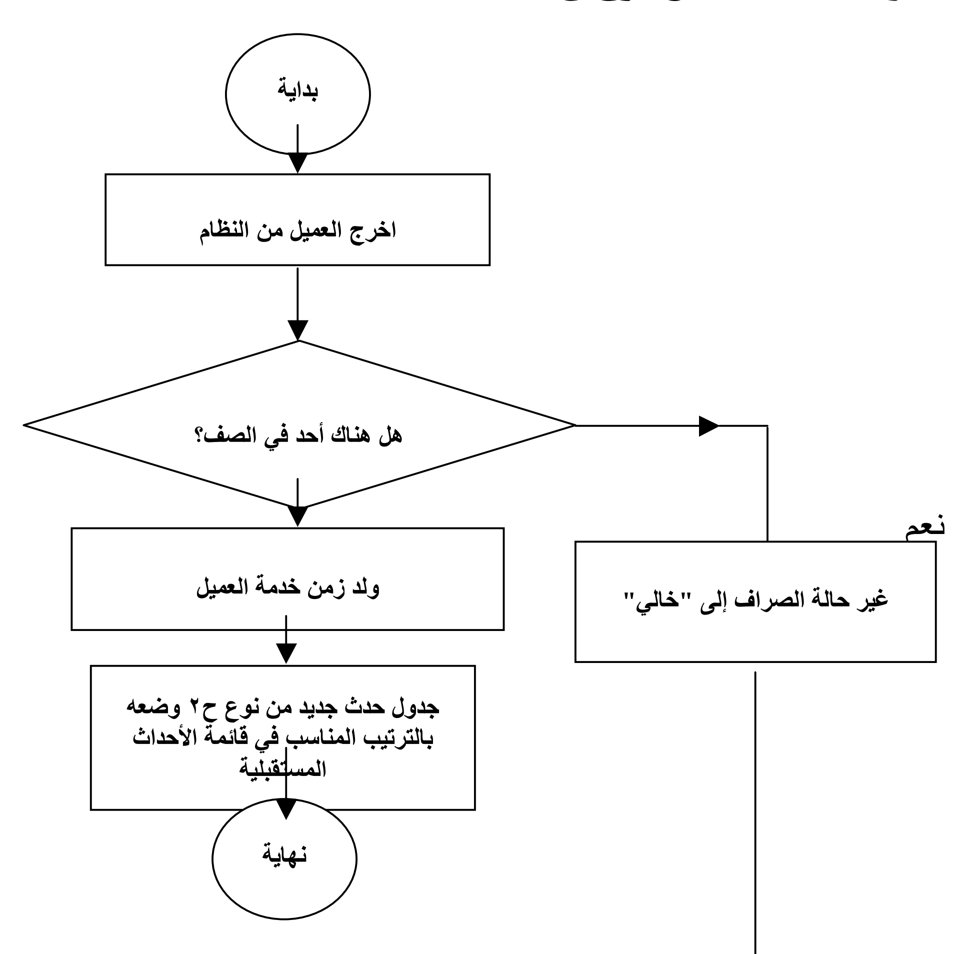
ح٢ : انتهاء خدمة عميل.

ولكل من هذين النوعين ، علينا كتابة الخارطة المنطقية التي توضح الإجراء الذي يجب تنفيذه لمحاكاة الأحداث من كل من هذين النوعين.

إجراء الأحداث من نوع ح١



إجراء الأحداث من نوع ح٢:



وبالإضافة إلى أسلوب قائمة الأحداث الذي سبق الحديث عنه في هذا الباب، هناك أساليب أخرى لتصميم برامج المحاكاة ، لعل من أهمها الأسلوب المبني على العمليات (Process-Based Approach) . وتتلخص فكرة هذا الأسلوب في جعل ما يسمى بالعملية (Process) هي وحدة البناء في النموذج . وتعرف العملية على أنها "سلسلة الخطوات التي تمر من خلالها

كل نوعية من الأشياء على مدى فترة حياتها في النظام" . ويورث الصنف الواحد أشياءه العلمية المرتبطة بهذا الصنف .

ويقوم البرنامج الرئيس في حال استخدام الأسلوب المبني على العمليات بمتابعة تقدم الأشياء البرمجية والتي تمثل كيانات النموذج في عملياتها ، ويقوم أيضاً بمحاولة دفعها للتقدم في إنجاز خطوات عملياتها بقدر المستطاع . وتتكون عملية كل صنف من الكيانات من مجموعة خطوات تمثل خطوات المعالجة التي يمر بها كل كيان؛ إلا أن بعض هذه الخطوات يتم عنده توقيف الكيان بصورة مؤقتة . وتنقسم نقاط التوقف هذه إلى نوعين (Pidd, 2000) .

١- نقاط التوقف غير المشروط :

وهي النقاط التي يحصل عندها توقف لتقدم الكيان في عمليته كفترة زمنية محددة مسبقاً (عن طريق أخذ عينة عشوائية من سلسلة أرقام عشوائية لتمثيل زمن التوقف) . ومن الأمثلة على هذه النوعية توقف العميل لتلقي الخدمة لدى الخادم وذلك في النماذج التي تمثل أنظمة صفوف الانتظار .

٢- نقاط التوقف المشروط :

وهي النقاط التي يحصل عندها توقف لتقدم الكيان في عمليته إلى حين تحقق شرط معين في نموذج المحاكاة . ومن الأمثلة على هذه النوعية توقف العميل في الصف إلى أن يأتيه الدور للخدمة عندما يتحقق الشرط بأن يكون في مقدمة الصف وأن تكون حالة الخادم هي "خالي" .

وتسمى النقاط التالية مباشرة لنقاط التوقف في عملية كل صنف من الكيانات بنقاط "إعادة التنشيط" (Re-activation Points). ويعتمد عمل

البرنامج الرئيس للمحاكاة بأسلوب العمليات على تدوين سجلات لكل كيان في المحاكاة تحتوي على زمن إعادة التنشيط لذلك الكيان (إذا كان التوقف غير مشروط) ، بالإضافة إلى تعريف موضع نقطة إعادة التنشيط في العملية . كما يتابع البرنامج الرئيس قائمتين من الأحداث هما :

أ- قائمة الأحداث المستقبلة .

ب- قائمة الأحداث الحالية .

أما قائمة الأحداث المستقبلة فتحوي قائمة من الأحداث التي تمثل أحداث إعادة تنشيط كل كيان تم توقيفه توقفاً غير مشروط . ويتم بالطبع ترتيب هذه الأحداث ترتيباً تصاعدياً حسب أزمنة إعادة التنشيط لها في نطاق زمن المحاكاة.

وأما قائمة الأحداث الحالية فتحوي قائمة أحداث إعادة التنشيط للكيانات التي كانت متوقفة توقفاً غير مشروط وحل زمن إعادة تنشيطها (أصبح زمن ساعة المحاكاة أكبر من أو يساوي زمن إعادة التنشيط). وبالإضافة لذلك، تحوي هذه القائمة أيضاً سجلات لكل الكيانات التي توقفت توقفاً مشروطاً إلى أن يتحقق الشرط الخاص بكل منها ، فيزال سجلها عندها من القائمة. ويعمل البرنامج الرئيس للمحاكاة المبنية على العمليات من خلال تكرار الخطوات التالية (Pidd, 2000):

١- مسح الأحداث المستقبلة:

وهنا يتم أخذ أول حدث على قائمة الأحداث المستقبلة ومساواة ساعة المحاكاة بزمن حصول هذا الحدث.

٢- تحريك الأحداث بين القائمتين:

وهنا يتم نقل الحدث الذي حل زمن حدوثه من قائمة الأحداث المستقبلة إلى قائمة الأحداث الحالية.

٣- مسح الأحداث الحالية:

وهنا يتم دفع الكيان الذي حل زمن إعادة تنشيطه ونقله إلى قائمة الأحداث الحالية بقدر المستطاع خلال عمليته حتى يصل إلى نقطة توقف جديدة. كما يتم أيضاً دفع الكيان الذي كان موقفاً توقفاً مشروطاً وتحقق شرط إعادة تنشيطه بقدر المستطاع في عمليته حتى يصل مرة أخرى إلى نقطة توقف جديدة.

ونقطة التوقف الجديدة ستتراوح بين نقطة توقف مشروط يبقى على إثرها سجل الكيان في قائمة الأحداث الحالية، أو نقطة توقف غير مشروط ينقل على إثرها سجل الكيان إلى قائمة الأحداث المستقبلة على شكل حدث إعادة التنشيط في المستقبل.

ويبقى أن نذكر أن الخيار بين أسلوب قائمة الأحداث والأسلوب المبني على العمليات لبناء برامج المحاكاة يعتمد على المحلل وخلفيته وعلى النظام وتفاصيله. وعموماً، فإن الأسلوب المبني على العمليات يميل إلى تبسيط النموذج على حساب زيادة صعوبة كتابة برنامج المحاكاة، على عكس أسلوب قائمة الأحداث الذي يكون بناء النموذج فيه مختلفاً وأقل سلاسة في مقابل سهولة برمجته كبرنامج محاكاة.

الباب الثالث

تصميم نماذج المحاكاة الإشهارية والوظيفية

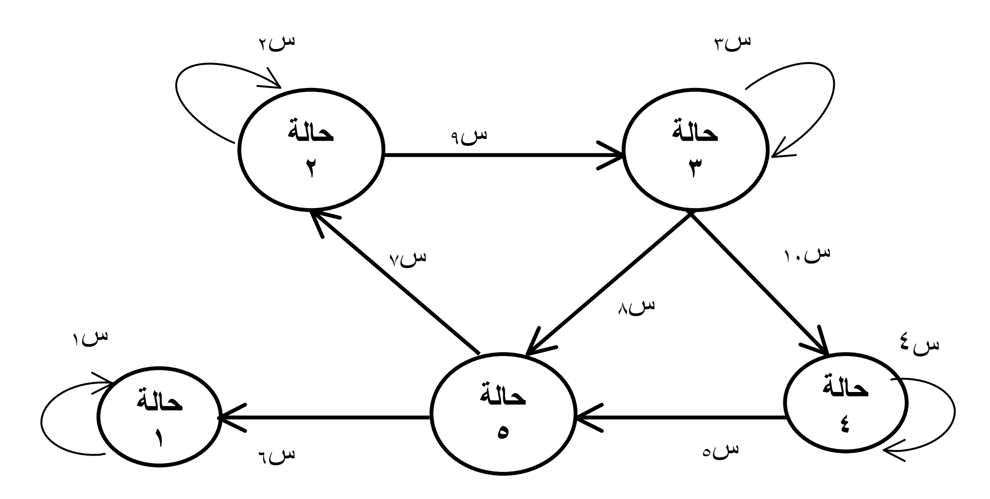
٣-١ المقدمة :

سنستعرض في هذا الباب كيفية تصميم نماذج المحاكاة بشيء من التفصيل لبعض أنواع نماذج المحاكاة الأكثر شيوعاً ، وهي النماذج الإشهارية (Declarative Models) والنماذج الوظيفية (Functional Models) . وسنقوم في سبيل ذلك بتعريف كل من النوعين المذكورين وأوجه ومواضع استخدامهما ، والخوارزميات المستخدمة لبناء النماذج وفقاً لكل نوع منهما . وسنعرض بعض الأمثلة لأنظمة يراد محاكاتها ونوضح كيفية بناء نماذج لها وفقاً لكل نوع من نوعي نماذج المحاكاة.

٣-٢ تصميم النماذج الإشهارية **(Declarative Models) :**

يعتمد الأسلوب الإشهاري (Declarative) للنمذجة على تصوير النظام من خلال سلسلة من الحالات التي يتقلب بينهما النظام، والتي تمثل مجموعة منتهية من النقاط في فراغ الحالة. وتستخدم النماذج الإشهارية عادة لنمذجة الأنظمة التي تعمل من خلال أطوار زمنية تمثل حالات محدودة يتقدم خلالها النظام عند وقوع الأحداث ، فبالتالي فهي تركز على جزئية حالات النظام وليس وظائف النظام وما يقوم به من معالجة للمدخلات لتحويلها إلى مخرجات. ولهذا السبب لا يحبذ استخدام النماذج الإشهارية لنمذجة الأنظمة التي ينظر إليها على أنها مكونة من مجموعة من المكونات الشيئية (Fishwick, 1995). بل إنه يفضل لتمثيل هذه الأنظمة استخدام النماذج الوظيفية التي تبرز دور مكونات النظام. وقد تستخدم النماذج الإشهارية في هذه الحالة لنمذجة عمل بعض هذه المكونات ، فتكون بذلك طبقة ثانية من النماذج بعد النماذج الوظيفية في تمثيل هذا النوع من الأنظمة .

وللشروع في بناء نموذج إشهاري للنظام ، فلابد من حصر وتعريف حالات هذا النظام والأحداث التي تؤدي إلى الانتقال من حالة لأخرى . ثم يتم تمثيل هذه الحالات – والتي يمكث النظام في كل منها لبرهة من الوقت – على شكل دوائر في شكل يسمى شكل انتقال الحالة (State – Transition Diagram) . وتمثل الأسهم التي تربط بين دوائر الحالات كيفية انتقال النظام من حالة إلى حالة أخرى عند وقوع حدث معين يرتبط بتغير قيمة مدخل أو أكثر من مدخلات النظام ، ويظهر هذا المدخل وقيمته التي تؤدي إلى انتقال حالة النظام أعلى السهم الذي يربط بين الحالة الحالية والحالة الجديدة. ويبين الشكل رقم (٣-١) مثالاً لنموذج إشهاري لأحد الأنظمة. وتسمى النماذج الإشهارية التي توصف من خلال أشكال انتقال الحالة النماذج المبنية على الحالة . وسنرى لاحقاً أنه بالإمكان أيضاً وصف النماذج الإشهارية عن طريق بناء النموذج الإشهاري اعتماداً على الأحداث التي تقع فيه بحيث تكون هذه الأحداث هي محور ارتكاز النموذج . ويسمى هذا النوع من النماذج الإشهارية النماذج المبنية على الأحداث.



الشكل رقم (٣-١) : شكل انتقال الحالة لوصف نموذج إشهاري لأحد الأنظمة بين عدد من الحالات اعتماداً على قيمة المدخل س = { $س_{١}$، $س_{٢}$، ... ، $س_{١٠}$}

وبالعودة إلى الشكل رقم (٣-١) ، نلاحظ أن انتقال النظام من كل حالة إلى الحالة المجاورة لها تم بناء على حصول قيمة معينة لمتغير المدخل س. ونشير هنا إلى أن انتقال النظام بين الحالات قد يحدث أيضاً بناء على حصول قيم معينة لمجموعة من متغيرات المدخلات وليس مدخلاً واحداً. وإذا كان انتقال النظام يتم قطعاً إذا ما حصلت القيم المعينة من متغيرات المدخلات فيكون النظام محدداً (Deterministic) . أما إذا كان انتقال النظام من حالة إلى حالة يحدث فقط باحتمالية إحصائية معينة فيكون النظام عشوائياً أو غير محدد (Nondeterministic) .

ولبناء نموذج محاكاة إشهاري ، علينا تحويل شكل انتقال الحالة للنظام إلى خوارزم يقبل البرمجة على الحاسب الآلي. ولعمل ذلك سنحتاج إلى تعريف متغيرات أساسية في برنامج المحاكاة تناسب طبيعة النماذج الإشهارية. على سبيل المثال ، سوف نحتاج إلى متغير يبين حالة النظام الحالة وقيمة أو قيم متغير أو متغيرات المدخلات للنظام ، كما سنحتاج إلى جدول يخزن قيم النطاق والنطاق المصاحب لما يسمى الحالة التحويلية للنظام والتي تقوم في هذه الحالة بإخراج معرف أو اسم الحالة الجديدة للنظام بدلالة معرف أو اسم الحالة الحالية وقيمة أو قيم متغيرات المدخلات الحالية ، وذلك لجميع حالات النظام ولجميع قيم متغيرات مدخلات النظام . فإذا توفر لدينا من شكل انتقال الحالة الجدول الكامل المعرف للدالة التحويلية

للنظام؛ فإنه يمكننا كتابة خوارزم محاكاة النظام حسب النموذج الإشهاري المحدد المبني على الحالة كالتالي (Fishwick, 1995):

١- ابتدئ قيم ساعة النظام عند الصفر وحالته عند الحالة الابتدائية .

٢- كرر حتى وقوع شرط نهاية المحاكاة ما يلي :

أ- اخرج معرف أو اسم الحالة الحالية للنظام .

ب- حرك ساعة المحاكاة بمقدار ثابت .

ج - انقل حالة النظام إلى الحالة التالية اعتماداً على الحالة الحالية وبناء على القيم الحالية لمتغيرات المدخلات .

<u>مثال:</u> صمم نموذج محاكاة لنظام إشارة مرور رباعية .

<u>الحل:</u> إذا افترضنا أن إشارة المرور في كل اتجاه تتغير بين الألوان أحمر (R) وأخضر (G) وأصفر (Y) .

ونفترض أن اتجاه السير يكون مفتوحاً في اتجاه واحد فقط في آن واحد ، أي أن الإشارة تكون على اللون الأخضر إما في الاتجاه الشمالي (N) أو الجنوبي (S) أو الشرقي (E) أو الغربي (W) . كما نفترض أن زمن بقاء الإشارة في كل من الاتجاهات الأربعة على اللون الأخضر هو (٣ز) وعلى اللون الأصفر هو (ز) .

ونخلص مما سبق أن نظام الإشارة الضوئية للتقاطع الرباعي سوف يمر بثماني حالات :

١- الحالة (NG) وتمثل الحالة عندما تكون الإشارة خضراء في الاتجاه الشمالي وحمراء في الاتجاهات الثلاثة الأخرى .

٢- الحالـة (NY) وتمـثل الحالـة عندما تكـون الإشارة صفراء في الاتجاه الشمالي وحمراء في الاتجاهات الثلاثة الأخرى .

٣- الحالـة (SG) وتمـثل الحالـة عندما تكون الإشارة خضراء في الاتجاه الجنوبي وحمراء في الاتجاهات الثلاثة الأخرى .

٤- الحالـة (SY) وتمـثل الحالـة عندما تكون الإشـارة صفراء في الاتجاه الجنوبي وحمراء في الاتجاهات الثلاثة الأخرى .

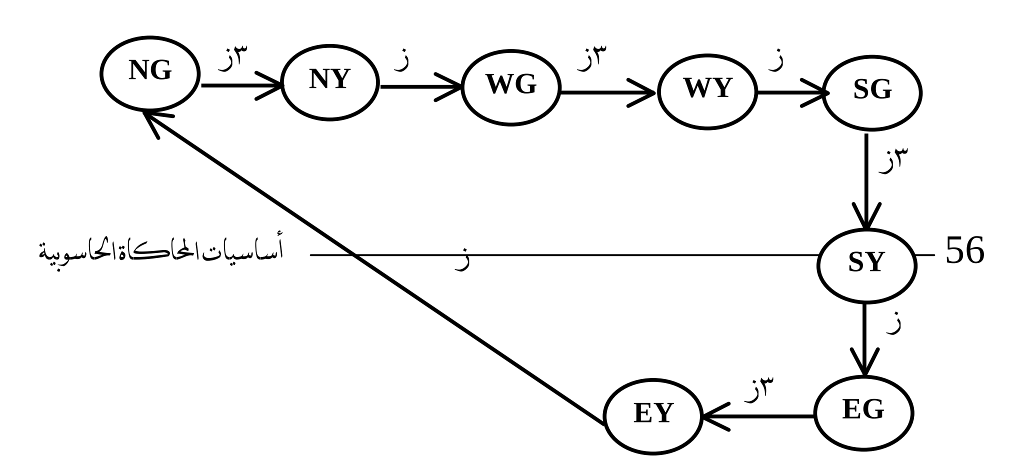
٥- الحالـة (EG) وتمـثل الحالـة عندما تكون الإشارة خضراء في الاتجاه الشرقي وحمراء في الاتجاهات الثلاثة الأخرى .

٦- الحالـة (EY) وتمـثل الحالـة عندما تكون الإشـارة صفراء في الاتجاه الشرقي وحمراء في الاتجاهات الثلاثة الأخرى .

٧- الحالـة (WG) وتمـثل الحالة عندما تكون الإشارة خضراء في الاتجاه الغربي وحمراء في الاتجاهات الثلاثة الأخرى .

٨- الحالـة (WY) وتمـثل الحالـة عندما تكون الإشارة صفراء في الاتجاه الغربـي وحمـراء فـي الاتجـاه الغربي وحمراء في ا لاتجاهات الثلاثة الأخرى .

وعليه فيكون تمثيل النظام على شكل نموذج إشهاري كما هو مبين في شكل انتقال الحالة في الشكل رقم (٣-٢) .



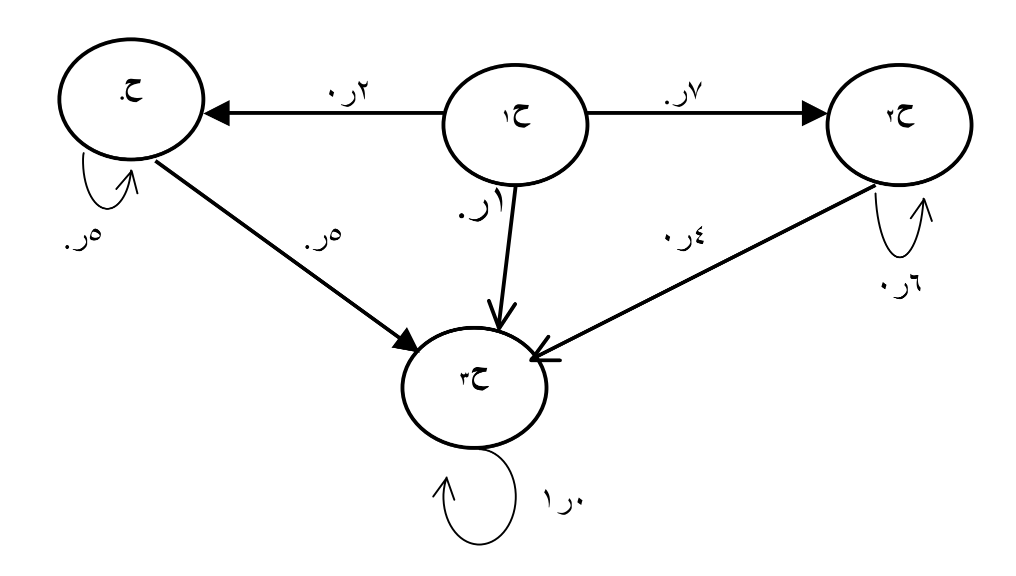
الشكل رقم (٣-٢) : شكل انتقال الحالة لنموذج إشهاري يحاكي عمل إشارة ضوئية لتقاطع رباعي

٣-٢-١ النماذج الإشهارية للأنظمة العشوائية:

تتميز الأنظمة العشوائية التي يمكن نمذجتها باستخدام النماذج الإشهارية بوجود عدد محدود من حالات النظام التي يتقلب النظام بينها بصورة عشوائية وباحتماليات معروفة للانتقال من حالة إلى حالة أخرى مجاورة لها . وتعتمد هذه الاحتماليات على الحالة الراهنة فقط التي يكون فيها النظام ، ولا تعتمد على أي من الحالات السابقة لها التي مر بها النظام قبل أن يصل إلى حالته الراهنة. وتسمى مثل هذه النماذج الإشهارية العشوائية بنماذج ماركوف (Markov Models) أو سلاسل ماركوف (Markov Chains) . وتعرف سلسلة ماركوف رياضياً على أنها العملية العشوائية المتقطعة $X=\{X_n:n=1,2,\ldots\}$ التي تحقق الشرط التالي :

$$\Pr\{X_{n+1} = k \mid X_0, X_1, \ldots, X_n\} = \Pr\{X_{n+1} = k \mid X_n\}$$

ومعنى هذا التعريف في سياق النماذج الإشهارية أنه في نماذج ماركوف ، سوف تعتمد احتمالية انتقال النظام إلى الحالة الجديدة k فقط على الحالة الحالية للنظام X_n ، ولن تعتمد على أي من الحالات التي مر بها النظام سابقاً قبل هذه الحالة. وعادة ما يتم تمثيل نماذج ماركوف على شكل مصفوفة تسمى مصفوفة الانتقال (Transition Matrix) يمثل كل عنصر P(i, j) فيها احتمالية انتقال النظام من الحالة i إلى الحالة j في هذا النموذج. ورسومياً، يمكن تمثيل نماذج ماركوف أيضاً من خلال شكل انتقال الحالة ، إلا أن الأسهم التي توصل بين حالات النظام المختلفة يتم ترقيمها بقيمة احتمالية انتقال النظام من الحالة الأولى إلى الحالة الثانية. ويوضح الشكل رقم (٣-٣) مثالاً لشكل انتقال الحالة لنموذج إشهاري عشوائي لأحد الأنظمة يوضح كيفية تمثيل حالات النظام المختلفة وعملية انتقال النظام من حالة إلى حالة حسب الاحتماليات المبينة على الأسهم الرابطة بين هذه الحالات .



الشكل رقم (٣-٣) : شكل انتقال الحالة لنموذج إشهاري عشوائي لأحد الأنظمة يبين احتماليات انتقال النظام من كل حالة إلى الحالة المجاورة لها

ونلاحظ في الشكل رقم (٣-٣) أن مجموع الاحتماليات للأسهم الخارجة من كل حالة يجب أن يكون واحداً صحيحاً ، نظراً لأن كل سهم خارج من كل حالة يعبر عن احتمال إحصائي لانتقال حالة النظام من الحالة التي يخرج منها السهم إلى الحالة التي ينتهي إليها السهم . كما نلاحظ أنه من الممكن أيضاً أن نمثل النموذج الذي يشير إليه الشكل رقم (٣-٣) على شكل مصفوفة انتقال كالتالي :

$$P= \begin{bmatrix} P(0,0) & P(0,1) & P(0,2) & P(0,3) \\ P(1,0) & P(1,1) & P(1,2) & P(1,3) \\ P(2,0) & P(2,1) & P(2,2) & P(2,3) \\ P(3,0) & P(3,1) & P(3,2) & P(3,3) \end{bmatrix}$$

$$P = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0.2 & 0 & 0.7 & 0.1 \\ 0 & 0 & 0.6 & 0.4 \\ 0.1 & 0 & 0 & 0 \end{bmatrix}$$

ومما سبق يتضح لنا أنه يلزمنا معرفة الحالات التي يتقلب بينها النظام العشوائي المطلوب بناء نموذج إشهاري له ، وأيضاً معرفة الاحتماليات

لانتقال هذا النظام من حالة إلى أخرى. ولو توفرت هذه المعلومات للمحلل فبإمكانه كتابة خوارزم محاكاة مثل هذا النظام بدلالة النموذج الإشهاري العشوائي (نموذج ماركوف) كالتالي (Fishwick, 1995):

١- ابتدئ قيم ساعة النظام عند الصفر وحالته عند الحالة الابتدائية .

٢- كرر حتى وقوع شرط نهاية المحاكاة ما يلي :

أ- اخرج معرف أو اسم الحالة الحالية للنظام .

ب- حرك ساعة المحاكاة إلى زمن وقوع حدث الانتقال إلى الحالة الجديدة.

ج- انقل حالة النظام إلى الحالة المجاورة الجديدة بناء على القيمة المسحوبة لمتغير احتمالية الانتقال العشوائي .

٣-٢-٢ النماذج الإشهارية غير العشوائية المبنية على الأحداث :

ذكرنا في معرض تعريفنا سابقاً لمفهوم الحدث والحالة أن الحدث هو مفهوم مرتبط بمفهوم الحالة من حيث إن الحدث هو اللحظة التي ينتقل فيها النظام من حالة سابقة - مكث فيها بعض الوقت- إلى حالة أخرى جديدة يمكث أيضاً فيها بعض الوقت.

ومن مقتضيات هذا الارتباط أنه يمكن بناء النماذج الإشهارية التي رأيناها حتى الآن بصورة مكافئة بحيث تبنى هذه النماذج على مفهوم "الحدث" بدلاً من "الحالة" وتسمى مثل هذه النماذج بالنماذج الإشهارية المبنية على الأحداث، وتمثل هذه النماذج باستخدام أشكال تسمى "رسوم الأحداث" أو (Event Graphs) والتي تمثل فيها الدوائر أحداثاً بدلاً من

ح١ ١٠ ح٢ ٤ ح٣

الحالات ، وتمثل فيها الأسهم الموصلة بين الدوائر الحالات التي يتواجد فيها النظام بين وقوع الحدث ووقوع الحدث الذي يليه ، كما يصاحب كل من هذه الأسهم بيان بطول الفترة الزمنية التي يمكث فيها النظام في هذه الحالة الحالية قبل وقوع الحدث التالي والذي يغيره إلى حالة أخرى. ويوضح الشكل رقم (٣-٤) مثالاً لرسم الأحداث لأحد الأنظمة يتكون من ستة أحداث متكررة هي {ح$_{١}$ ، ح$_{٢}$، ح$_{٣}$، ح$_{٤}$، ح$_{٥}$ ، ح$_{٦}$} :

الشكل رقم (٣-٤) : رسم الأحداث لنظام يمكن بناء نموذج إشهاري له من ستة أحداث

ونرى من الشكل رقم (٣-٤) أن النظام المبين في الشكل يعمل من خلال وقوع سلسلة متكررة من الأحداث هي ح$_{١}$ والتي تضعه في حالة يمكث بها لمدة ١٠ وحدات زمنية، ثم ينتقل بعدها على حالة أخرى بوقوع الحدث ح$_{٢}$ يمكث بها ٤ وحدات زمنية ، ثم يقع الحدث ح$_{٣}$ فينتقل إلى الحالة التالية والتي يمكث بها لمدة ٧ وحدات زمنية ، وهكذا حتى يقع الحدث ح$_{٦}$ والذي يمكث بعده النظام في حالة جديدة لمدة ٥ وحدات زمنية ثم يقع بعدها مرة أخرى الحدث الأول في الدورة ح$_{١}$.

ونظراً للارتباط المنطقي بين النماذج الإشهارية المبنية على الحالات وتلك المبنية على الأحداث، فإنه بالإمكان التحويل من أحد النوعين إلى الآخر. فللتحويل من نماذج مبنية على الحالة على نماذج مبنية على الأحداث علينا القيام بالتالي :

١- استبدال أسماء الحالات في شكل انتقال الحالة بأسماء الأحداث.

٢- استبدال قيم المدخلات على الأسهم في شكل انتقال الحالة بقيم الأزمنة التي يمكث فيها النظام في كل من الحالات التي يمر بها.

وبهذه الكيفية نحصل على رسم الأحداث المحدد للنموذج المبني على الأحداث من شكل انتقال الحالة المعبر عن النموذج المبني على الحالة. وللتحويل من رسم الأحداث إلى شكل انتقال الحالة ، علينا عمل خطوات مماثلة كالتالي:

١- استبدال الأحداث المتتالية في رسم الأحداث بأسماء للحالات التي يكون عليها النظام بين كل من هذه الأحداث.

٢- استبدال الأزمنة على الأسهم الموصلة بين الأحداث في رسم الأحداث بقيم جديدة لواحد أو مجموعة من مدخلات النظام الخارجية أو الداخلية والتي يعبر تغير هذه المدخلات إليها عن وقوع حدث معين يؤدي إلى تغير حالة النظام من الحالة الراهنة إلى الحالة التالية في شكل انتقال الحالة . ويقتضي ذلك تعريف متغيرات المدخلات (الداخلية للنظام أو الخارجية عن النظام) المناسبة للتعبير عن أحداث النظام المختلفة.

٣-٣ تصميم النماذج الوظيفية (Functional Models) :

تستخدم النماذج الوظيفية عندما يتكون النظام المطلوب محاكاته من مجموعة أشياء (Objects) تتفاعل بداخل النظام مع بعضها البعض ولكل منها وظيفة محددة ذات خصائص واضحة . وعند بناء النموذج الشيئي للنظام (Object-Oriented Model) تمثل هذه الوظائف على شكل طرق (Methods) وتتحول الخصائص إلى صفات (Attributes) وعندما تكون هذه الأشياء متصلة ببعضها بشبكة أو مسارات محددة ، فعندئذ يُعرف هذا النموذج بالنموذج الوظيفي. وتتميز الأنظمة التي تتوافق مع النمذجة الوظيفية بوجود تدفق لكائنات مؤقتة من مكونات النظام بين الأشياء المستديمة التي توجد في هذا النظام، وبالترتيب أو المسار الذي تحدده الشبكة الرابط بين هذه الأشياء.

ومن أمثلة هذه الأنظمة في الواقع ما يعرف بأنظمة صفوف الانتظار (Queuing Systems) التي تتكون من محطات معالجة هي بمثابة أشياء مستديمة تقوم بوظيفة معينة على كل ما يمر بها من كائنات مؤقتة تسري في النظام ، ولكل منها خصائص معينة تميزها عن غيرها. ويكون مرور الكائنات المؤقتة عبر المحطات من خلال مسار محدد تعبر عنه الشبكة الرابطة بين هذه المحطات. ومن أمثلة أنظمة صفوف الانتظار في الواقع أنظمة المصانع التي تمر فيها المصنوعات عبر آلات معينة تمثل محطات معالجة وبترتيب ومسار معين حتى ينتهي تصنيفها وتخرج باعتبارها منتجًا

متكاملاً. وفيما يلي نلقي الضوء على تفاصيل أنظمة صفوف الانتظار وكيفية بناء النماذج الوظيفية.

٣-٤ أنظمة طوابير الانتظار **(Queuing Systems) :**

كما أوردنا سلفاً ، تستخدم النماذج الوظيفية لنمذجة الأنظمة التي تتكون من وحدات بناء مستقلة تقوم كل منها بوظيفة محددة وترتبط مع بعضها البعض في شبكة ذات هيكل واتجاه معين . وتقوم كل من هذه الوحدات بتنفيذ عملية تمثل الوظيفة المميزة لهذه الوحدة تقوم من خلالها بمعالجة الكائنات المؤقتة التي تمر خلالها وتقديم *الخدمة* لها . وينتج عن ذلك تغيير في قيمة متغيرات الحالة لهذه الكائنات المؤقتة التي تغادر وحدة المعالجة الحالية إلى المحطة التالية في شبكة الوحدات التي يتكون منها مثل هذا النظام . ويسمى مثل هذا النظام بنظام طوابير الانتظار (Queuing System) ولنمذجة مثل هذا النظام سوف نحتاج إلى تمثيل أحداث وصول الأشياء المؤقتة إلى محطات الخدمة ، وحصول هذه الأشياء على الخدمة ثم مغادرتها ، بالإضافة إلى تمثيل انتقال الأشياء المؤقتة (ويطلق عليها أيضاً العملاء) من طابور إلى طابور حتى انتهاء خدمتهم وخروجهم من النظام .

وتتكون أنظمة طوابير الانتظار من ثلاثة عناصر رئيسة تكتسب منهم خصائصها الأساسية وهم:

١- عملية وصول العملاء .

٢- عملية خدمة العملاء .

٣ـ نظام الطابور .

وتصف عملية وصول العملاء إلى النظام الكيفية التي يصل بها العملاء إلى النظام من الناحية الإحصائية ، حيث تحدد عملية الوصول من خلال تحديد التوزيع الإحصائي للمتغيرات العشوائية .... $A_1, A_2, A_3,$ التي تمثل الأزمنة الفاصلة بين وصول العميلين رقم ١،٢ ، وبين العميلين ٢ ، ٣ وهكذا . وعادة ما يفترض ـ بناء على خصائص النظام ـ أن التوزيع الإحصائي متماثل لهذا المتغيرات العشوائية وأنها مستقلة إحصائياً . أما عملية خدمة العملاء فيحددها عدد محطات الخدمة ونوعية التوزيع الإحصائي للمتغير العشوائي الذي يمثل زمن الخدمة عند كل من هذه المحطات. وفيما يتعلق بنظام الطابور (Queuing Discipline) فهو بمثابة القانون الذي يحكم العملاء في طابور معين من حيث الأحقية في الدور للوصول إلى محطة الخدمة التي تلي هذا الطابور. فقد تكون الأولوية لمن وصل إلى الطابور أولاً أو لمن وصله آخراً أو لفئة معينة من العملاء، وهكذا.

وقد جرت العادة بتسمية نظام الطوابير بناء على نظام الرموز التالي : $L_1/L_2/m$ ، حيث يرمز الحرف L_1 إلى نوع التوزيع الإحصائي للمتغيرات العشوائية للأزمنة الفاصلة بين وصول العملاء ، ويرمز الحرف L_2 إلى نوع التوزيع الإحصائي للمتغير الإحصائي الذي يمثل عملية خدمة العملاء في النظام، كما يمثل العدد الصحيح m عدد محطات الخدمة التي تلي طابور الانتظار الجاري وصفه. فعلي سبيل المثال، عندما يكون التوزيع الإحصائي لمتغيرات وصول العملاء من نوع (Exponential)، وعندما يكون التوزيع الإحصائي لمتغير خدمة العملاء أيضاً من نوع (Exponential) وعدد

محطات الخدمة هو واحد ، فإن نظام طابور الانتظار يرمز له بالرمز (M/M/l) ، حيث يستخدم الحرف M للإشارة إلى توزيع (Exponential) نسبة إلى خاصية الماركوفية المعروفة لهذا التوزيع في علم الإحصاء.

ومن أهم الكميات التي تصف خصائص أنظمة الطوابير ما يلي :

١ـ متوسط زمن خدمة العميل (S) .

٢ـ متوسط زمن انتظار العميل في الطابور(D) .

٣ـ المتوسط في وحدة الزمن لعدد العملاء في الطابور(Q) .

٤ـ معدل وصول العملاء إلى النظام (λ).

٥ـ معدل خدمة العملاء في محطة الخدمة ($1/\mu$) .

٦ـ نسبة الاستفادة من محطة الخدمة (Server Utilization) التي يرمز لها عادة بالرمز (ρ) وتحسب باستخدام العلاقة التالية :

$$r = \frac{l}{m}$$

ويمكن بالرجوع إلى نظرية أنظمة الطوابير إيجاد صيغ مغلقة لحاسب كميات تعبر عن مقاييس أداء أنظمة طوابير الانتظار مثل :

$$Q = \lambda\ D$$

$$L = \lambda W$$

حيث L هي المتوسط الزمني لعدد العملاء في النظام (تساوي Q + عدد العملاء الجاري خدمتهم) وW هي متوسط زمن انتظار العميل في

النظام متوسط زمن الانتظار في الطابور وأثناء الخدمة. وعلى الرغم من وجود علاقات رياضية محددة تصف خصائص أنظمة الطوابير من خلال تحديد قيمة متغيرات مقاييس الأداء المهمة لهذه الأنظمة ، إلا أن هذه الصيغ يمكن اشتقاقها فقط لبعض تلك الأنظمة التي لها خصائص إحصائية معينة يمكن معها اشتقاق هذه الصيغ الرياضية بنجاح . أما ما عدا هذه الأنظمة ، فلا تسعف نظرية طوابير الانتظار الرياضية في حساب خصائصها وينبغي لعمل ذلك الاعتماد على وسائل أخرى بديلة مثل المحاكاة الحاسوبية .

ولمحاكاة أنظمة طوابير الانتظار المتعددة المحطات ، ينبغي أولاً التعرف إلى أسلوب نمذجة طابور الانتظار البسيط الأحادي الخدمة . وإذا اعتمدنا أسلوب المحاكاة المتقطعة الأحداث فيمكن تمثيل نموذج محاكاة طابور الانتظار الأحادي الخدمة وفقاً للخوارزم التالي :

١- ابتدئ قيم ساعة النظام عند الصفر وحالته عند الحالة الابتدائية لجميع المحطات .

٢- كرر حتى وقوع شرط نهاية المحاكاة ما يلي:

أ- إذا كان الحدث الحالي هو "وصول عميل" نفذ إجراء محاكاة حدث "وصول عميل" .

ب- إذا كان الحدث الحالي هو " انتهاء خدمة عميل" نفذ إجراء محاكاة حدث "إنهاء خدمة عميل".

٣- اطبع النتائج الإحصائية للمحاكاة.

ومن الممكن كتابة إجراء حدث " وصول عميل" على النحو التالي :

أ- جدول حدث وصول عميل جديد.

ب- إذا كانت محطة الخدمة شاغرة ولا يوجد أحد في الطابور ابدأ الخدمة وجدول حدث "انتهاء خدمة عميل".

وإذا كان هناك أحد في الطابور ، أضف العميل الجديد إلى نهاية الطابور.

كما يمكن كتابة إجراء حدث "انتهاء خدمة عميل" على النحو التالي:

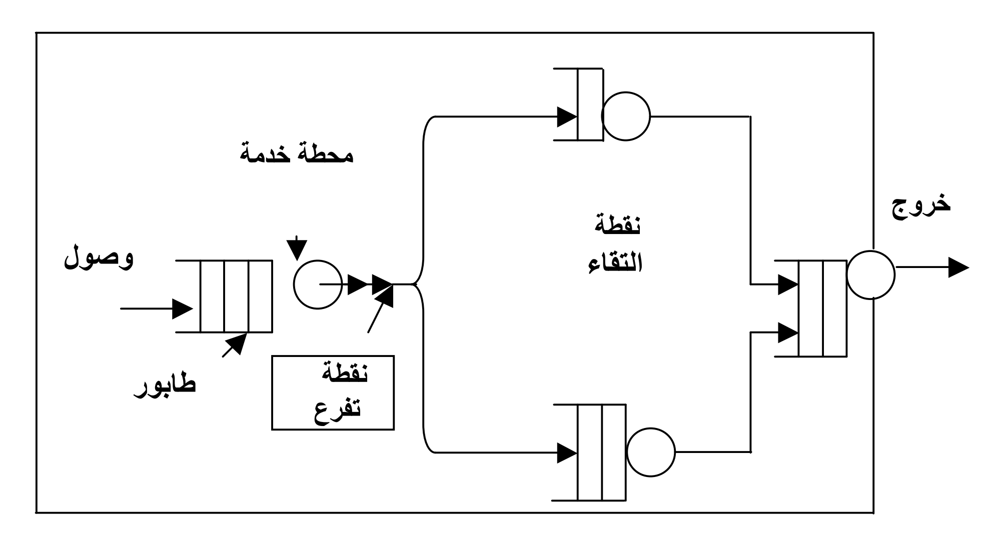
أ- احذف العميل من النظام.

ب- إذا كان هناك عملاء ينتظرون في الطابور ابدأ خدمة العميل التالي في الدور وجدول حدث جديد من نوع " انتهاء خدمة عميل "

وإذا لم يكن هناك عملاء ينتظرون في الطابور أعلن أن محطة الخدمة شاغرة .

وعموماً ، من المرجح أن يتألف نظام طوابير الانتظار من مجموعة متشابكة من محطات الانتظار التي يسبق كل منها طابور(أو يسبق كل مجموعة منها طابور) ، ويسري العملاء منذ لحظة دخولهم عبر هذه الشبكة في اتجاه محدد حتى يتموا حصولهم على الخدمة ثم يغادروا حدود النظام. ومن المحتمل هنا أن تكون حركة العملاء بعد انتهاء الحصول على الخدمة من محطة مرحلية إلى المحطة التالية معتمدة على شرط معين بحسب نوع

العميل أو بحسب احتمالية إحصائية معينة. ويوضح الشكل رقم (٣-٥) مثالاً لشبكة من طوابير الانتظار تحتوي على نقاط تفرع كما هو مذكور أعلاه .



الشكل رقم (٣-٥) : مثال على شبكة من طوابير الانتظار ذات نقاط تفرع

مثال :

ارسم خرائط منطقية لنمذجة ومحاكاة نظام صراف بنكي باستخدام طوابير الانتظار مكون من موظف واحد يقوم بأعمال الصراف وصف انتظار واحد لعملاء البنك للوقوف لانتظار الخدمة . افترض أن معدل وصول العملاء إلى البنك هو (ص) عميل في الدقيقة ومتوسط زمن خدمة العميل الواحد هو (خ) دقيقة .

الحل :

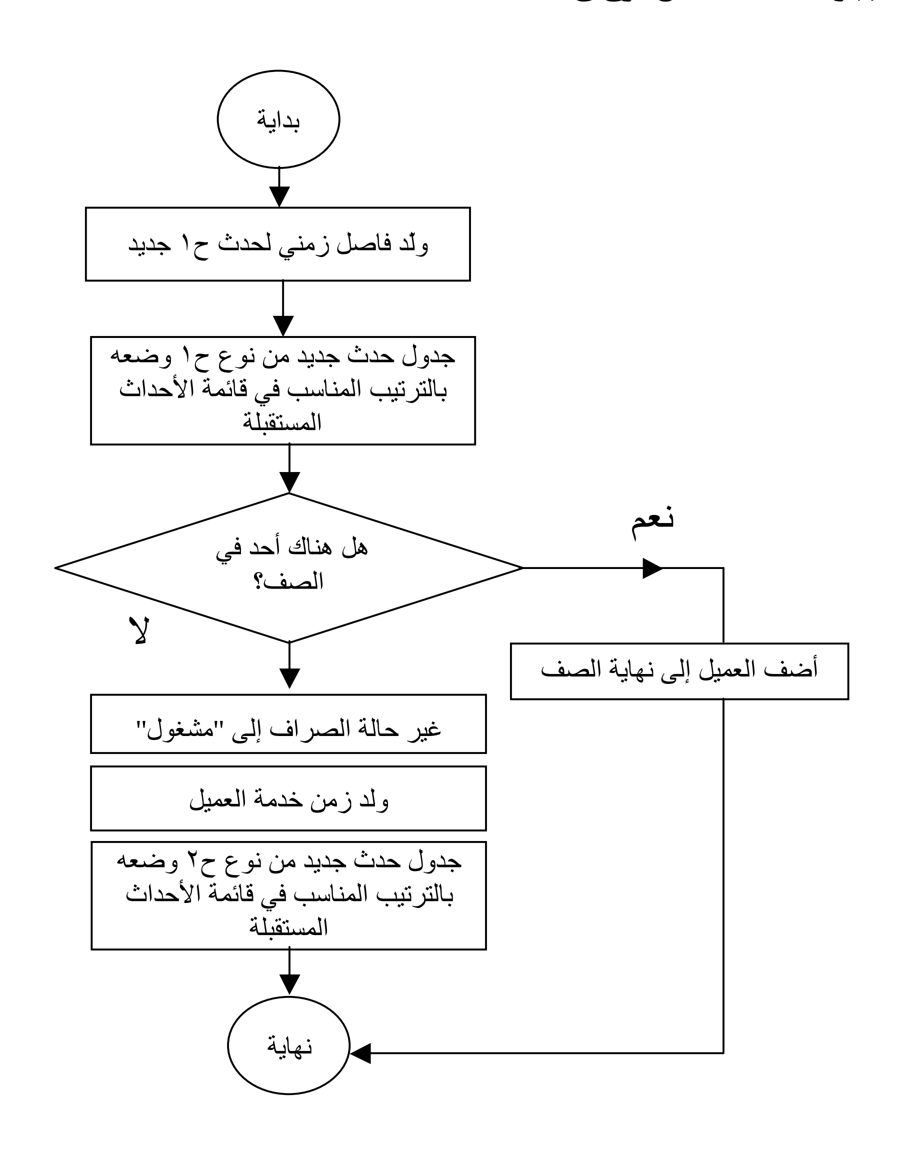
في هذا النظام ، يمكننا التعرف إلى النوعين التاليين من الأحداث :

ح١ : وصول عميل .

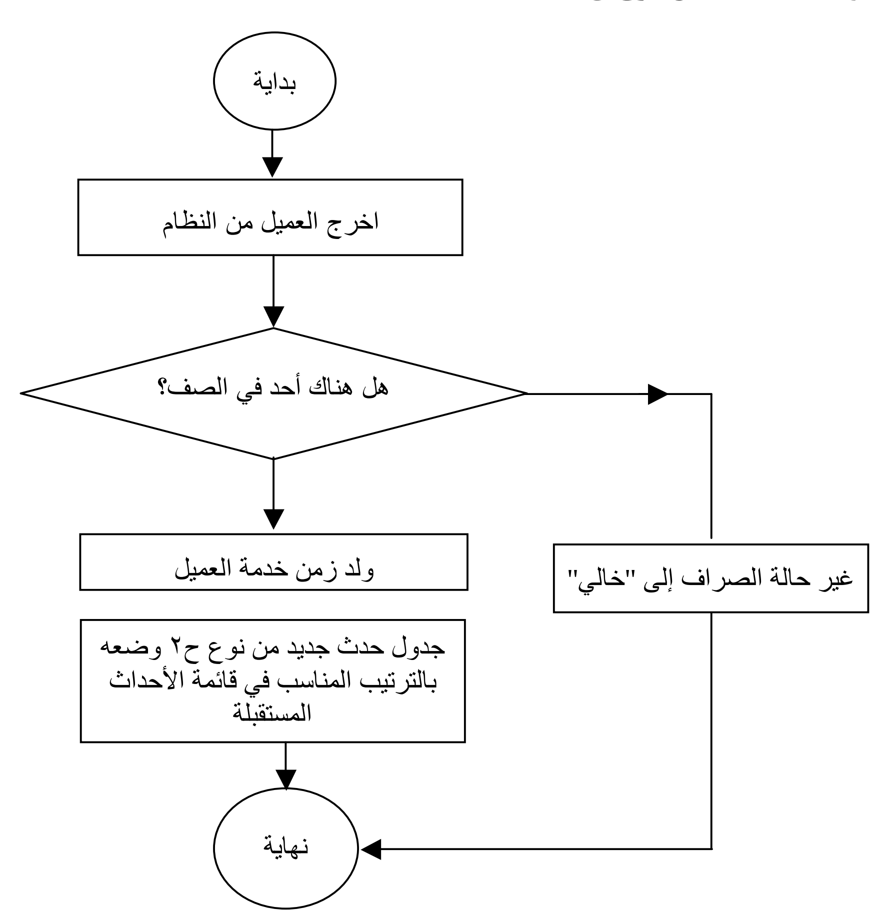
ح٢ : انتهاء خدمة عميل.

ولكل من هذين النوعين ، علينا كتابة الخارطة المنطقية التي توضح الإجراء الذي يجب تنفيذه لمحاكاة الأحداث من كل من هذين النوعين.

إجراء الأحداث من نوع ح١



إجراء الأحداث من نوع ح٢:



الباب الرابع

أدوات بناء برامج المحاكاة

٤-١ المقدمة :

سنناقش في هذا الباب برمجيات تطوير نماذج المحاكاة وفي هذا الصدد سينصب حديثنا على تصنيف هذه البرمجيات والفروق بينها، ومعايير التقييم والمفاضلة بين هذه البرمجيات. بعد ذلك سنأخذ واحدة من الحزم الحديثة لتطوير نماذج المحاكاة كحالة دراسية ونستعرض خصائصها وإمكاناتها بشيء من التفصيل.

٤-٢ أدوات بناء برامج المحاكاة :

تنقسم الأدوات البرمجية السائدة لتطوير برامج المحاكاة إلى ثلاثة أقسام:

١- لغات البرمجة الاعتيادية.

٢- لغات برمجة المحاكاة.

٣- بيئات المحاكاة (حزم تطوير برامج المحاكاة).

ويعتبر الصنف الأول (لغات البرمجة الاعتيادية) الأقل استخداماً من الناحية العملية كوسيلة لتطوير برامج المحاكاة، حيث إن هذه اللغات غير مصممة أصلاً لبرمجة نماذج المحاكاة وتفتقر بالتالي إلى التراكيب الضرورية لبناء تلك النماذج والتي لابد أن يبنيها المبرمج بنفسه (أو في أفضل الأحوال بالإستعانة ببعض المكتبات البرمجية المتخصصة في المحاكاة)؛ لتصبح بنية تحتية تمكنه من البدء في كتابة برنامج المحاكاة الأصلي. ومن أمثلة لغات البرمجة العامة التي قد تستخدم في تطوير برامج المحاكاة لغات (C++) و(Java) .

أما الصنف الثاني فهو لغات برمجة المحاكاة، وهي لغات برمجة مصممة خصيصاً لتطوير برامج المحاكاة وذلك بتوفير إطار برمجي عام يتوافق مع متطلبات برامج المحاكاة وصيغة برمجية (Syntax) مبسط يتمشى مع مصطلحات ومتطلبات برامج المحاكاة. وتتميز كل لغة محاكاة عن الأخرى من حيث كيفية تمثيلها لخصائص النموذج الثابتة (Static) وكيفية تمثيلها لخصائص النموذج الحركية (Dynamic). ويقصد بالخصائص الثابتة كيفية تعريف اللغة للكيانات والأشياء المكونة للنموذج والعلاقات بينها. ويقصد بالخصائص الحركية طريقة تحريك زمن المحاكاة للأمام ومتابعة الأحداث في اللغة. ومن أمثلة لغات المحاكاة الواسعة الانتشار لغة (GPSS) التي أنتجتها شركة (IBM) منذ الستينات الميلادية، ولا تزال النسخة المحدثة منها (GPSS/H) رهن الاستخدام حتى الآن. ومن اللغات الشهيرة الأخرى لغة (SIMSCRIPT) والتي تعتمد أسلوب قائمة الأحداث (Event List) أساساً لها، ولغة (SIMAN) الواسعة الانتشار خاصة في المجال الصناعي والتي تعتمد على أسلوب قائمة الأحداث لتشغيل نماذج المحاكاة.

ومن المزايا التي يوفرها الاعتماد على لغات برمجة المحاكاة وجود واجهة رسومية تمكن المبرمج من تعريف النموذج بشكل رسومي وتقديم العون له في ذلك. كما أن هذه اللغات توفر أدوات لتتبع واكتشاف الأخطاء البرمجية في شفرة البرنامج (Debuggers) . كما تقدم لغات برمجة المحاكاة العون للمبرمج أيضًا في توليد سلاسل الأرقام العشوائية من مجموعة واسعة ومتنوعة من التوزيعات الإحصائية. كذلك فإنها تعين المبرمج في عملية جمع البيانات الإحصائية الخارجة من برنامج المحاكاة أثناء التشغيل عن

طريق تحديد المتغيرات المطلوب تجميع قيمها أثناء تشغيل البرنامج وحساب الخصائص الإحصائية لها في نهايته بصورة آلية. كما توفر أغلب لغات برمجة المحاكاة إمكانية رؤية البيانات الإحصائية وكيفية تغيرها أثناء تشغيل برنامج المحاكاة بصورة رسومية على هيئة منحنيات أو خرائط (Charts) تتغير مع الزمن، كما أنها غالباً أيضاً ما توفر إمكانية تحريك أيقونات (Icons) تمثل كيانات النموذج أثناء تشغيل البرنامج لتعبر عن سلوكات النموذج بأسلوب الرسوم المتحركة أو ما يعرف باسم (Model Animation) .

أما الصنف الثالث من أدوات بناء برامج المحاكاة فهو الحزم البرمجية (Software Packages) والتي توفر بيئة متكاملة لتطوير برنامج المحاكاة. وعادة ما تكون هذه الحزم تفاعلية (Interactive) ومرئية (Visual) حيث إنها تعطي المحلل إمكانية تمثيل النموذج كاملاً بصورة مرئية (عادة بدون الحاجة إلى كتابة أي شفرة برمجة)، كما تعطيه إمكانية مشاهدة ما يحدث بداخله أثناء التشغيل بشكل رسومي متحرك (Animation) . وتمكن هذه الحزم المحلل أيضاً من التفاعل مع البرنامج من خلال إيقافه مؤقتاً وتعديل بعض معطياته أو مكوناته أو قيم مؤثراته ثم إعادته إلى حالة التشغيل مرة أخرى من النقطة الزمنية التي تم إيقافه عندها. ومن أبرز الإضافات التي توفرها هذه الحزم أيضًا البرامج التي تأتي عادة ملحقة بها للمساعدة في إجراء تجارب المحاكاة وتحليل مخرجاتها إحصائيًا وحساب فترات الثقة الإحصائية لها، وكذلك المعاونة في تصميم التجارب الإحصائية اللازمة للبحث عن القيم المثلى لمقاييس أداء النموذج المختارة. وتتفاوت حزم المحاكاة في طريقة تمثيلها لنماذج المحاكاة وطريقة تشغيلها لها، فبعضها

يعتمد أسلوب قوائم الأحداث وبعضها يعتمد أسلوب تفاعل العمليات، وإن كان أسلوب تفاعل العمليات هو الأكثر انتشاراً. ومن أمثلة حزم تطوير برامج المحاكاة الحديثة والشهيرة ما يلي:

١- حزمة (Arena) من شركة (Rockwell) .

٢- حزمة (ProModel) من شركة (PROMODEL).

٣- حزمة (Extend) من شركة (Imagine That, Inc) .

وسوف نقوم في الفصل التالي باستعراض عام لحزمة (Arena) كحالة دراسية نقوم من خلالها بالتعرف عن قرب إلى ملامح هذه الحزمة ومجالات استخدامها في محاكاة الأنظمة، وأسلوب بناء نماذج المحاكاة فيها، وما تقدمه من خواص وإمكانات دعم مختلفة للمحلل.

٤-٣ حزمة أرينا (Arena) لبناء نماذج المحاكاة :

حزمة أرينا من الحزم التي تمكن من محاكاة الأنظمة بأنواعها المختلفة المتقطعة منها والمتصلة. وتأتي أرينا في عدة إصدارات: (Basic) و (Standard) و (Professional) . وتعتبر أرينا (Standard) هي الطبعة الأساسية المصممة لبناء نماذج عامة لطائفة كبيرة من الأنظمة. أما طبعة (Basic) فهي لاستخدام المنشآت والمستخدمين ذوي الأنظمة الصغيرة، وبالتالي فهي تقدم جزءاً بسيطاً من إمكانات أرينا (Standard). وأما طبعة (Professional) فهي تزيد على أرينا الأصلية بقدراتها على استحداث مكونات وأطر جديدة للمحاكاة يكيفها المحلل حسب متطلبات النظام الأصلي ويقوم ببرمجة تفاصيلها بنفسه. وتعتمد حزمة أرينا على لغة المحاكاة (SIMAN)،

وتوفر خصائص عدة مساعدة للمحلل في بناء وتشغيل نماذج المحاكاة مثل الرسوم المتحركة واستيراد الرسوم وتصديرها للحزم الأخرى، كما توفر مؤشرات رسومية لأداء النظام يمكن عرضها لإعطاء انطباع فوري عن كيفية تغير حالة النموذج أثناء تشغيله.

وتشتمل حزمة أرينا أيضًا على عدة برامج ملحقة لدعم المستخدم في مختلف مراحل بناء نموذج المحاكاة، مثل برنامج (Input Analyzer) لميكنة عملية اختيار التوزيع الإحصائي والقيم الإحصائية المناسبة للبيانات الميدانية، وبرنامج (OptQuest) لمساعدة المستخدم في تصميم تجارب المحاكاة بهدف إيجاد القيم العظمى أو الصغرى لدالة أحد مخرجات المحاكاة تحت عدد من المحددات على قيم مؤثرات النظام ومتغيراته وبعدة طرق رياضية ومعروفة. كما تشتمل أرينا أيضاً على برنامج (Output Analyzer) لميكنة حساب فترات الثقة الإحصائية على مخرجات المحاكاة أو إجراء المقارنات بين مخرجات برنامج المحاكاة تحت التصميمات المختلفة مع الاستعانة بالمنحنيات المختلفة لتمثيل هذه المخرجات. وهناك أيضاً برنامج (Process Analyzer) لميكنة وإدارة عملية تعريف الحالات والسيناريوهات المطلوب دراستها في تجارب المحاكاة للوصول إلى هدف معين، مع إمكانية ترتيب البدائل التي تمثلها السيناريوهات المختلفة بحسب تفوق مخرجات المحاكاة الخاصة بها حسب المعايير التي يضعها المحلل. وفيما يلي سنقوم باستعراض أهم خصائص أرينا بشكل أكثر تفصيلاً مع عرض أمثلة توضيحية لهذه الخصائص.

٤-٣-١ لمحات عن حزمة أرينا (Arena) لتطوير برامج المحاكاة :

تتكون برامج المحاكاة المطورة باستخدام حزمة أرينا من وحدات بناء تسمى الوحدات (Modules) هي عبارة عن مكونات إما أن تختص بتعريف التسلسل المنطقي للبرنامج (Logical Flow) أو تختص بتعريف البيانات (Kelton, Sadowski and Sadowski, 2002) وتعتمد أرينا في تمثيل نماذج المحاكاة على الأسلوب المبني على العمليات، فبالتالي تعرف العملية الواحدة كسلسلة من الوحدات المنطقية ومجموعة من وحدات البيانات المصاحبة لها. ويتم اختيار أنواع الوحدات المناسبة للنموذج من مجموعة كبيرة من الوحدات التي تتوفر للمحلل من خلال عدد من الألواح (Panels) التي يتخصص كل منها في نوعية معينة من الوحدات، ويقوم المحلل باختيار ما يحتاج من الوحدات من اللوح المناسب وسومياً عن طريق سحب هذه الوحدة وإسقاطها في المكان المناسب في منطقة التصميم بأسلوب السحب والإسقاط (Drag and Drop). وتصف وحدات التسلسل المنطقي مراحل معالجة كيانات النموذج والتي تسري خلالها هذه الكيانات، فتتلقى فيها المعالجة التي يمليها النموذج أو تتعرض للإيقاف لانتظار تحقق شرط معين في النموذج ثم تستمر بعده في السريان حتى تصل إلى نهاية العملية فتغادر عندها النموذج. وأما وحدات البيانات فهي تعرف خصائص العناصر التي تتكون منها العمليات مثل الكيانات (Entities) والموارد (Resources) والصفوف (Queues). كما أنه يمكن من خلالها تعريف متغيرات وثوابت عامة للنموذج ككل. وتظهر هذه الوحدات على الشاشة كجداول من نوع (Spreadsheet) يقوم المحلل بتعبئتها بقيم المتغيرات والخصائص العددية لما يحتويه النموذج من وحدات منطقية.

وفيما يلي عرض ملخص لبعض خصائص وإمكانات حزمة أرينا:

- تحتوي أرينا على لوح (Basic) والذي يضم وحدات التسلسل المنطقي الأساسية مثل: (Create) و (Dispose) و (Process) و(Decide) و (Assign). كما تحتوي على وحدات البيانات الأساسية مثل (Entity) و (Queue)و(Resource) و(Variable) و (Schedule) و (Set).

- تستخدم وحدة (Create) باعتبارها مصدر توليد للكيانات المؤقتة بالنموذج، فبالتالي يمكن من خلال البيانات المصاحبة لهذه الوحدة تحديد معدل وخصائص عملية وصول الكيانات إلى النموذج، مثل التوزيع الإحصائي للزمن الفاصل بين كل وصول ووصول، وعدد الكيانات التي تصل سوياً في الوصول الواحد وأنواع هذه الكيانات. ونلاحظ هنا أنه يمكن إدخال هذه البيانات بواسطة تعبئة حقول تظهر على الشاشة عند الضغط المتكرر على أيقونة (Icon) وحدة (Create) في نافذة النموذج، أو من خلال تعبئة جدول يظهر في نافذة خاصة عند الضغط مرة واحدة على أيقونة (Create).

- تستخدم وحدة (Entity) لتعريف أنواع الكيانات الموجودة في النموذج والخصائص المصاحبة لها (مثل التكلفة) وتحديد ما إذا كان جمع الإحصاءات عن كيانات كل من هذه الأنواع مطلوباً أم لا، وتحديد أيضاً شكل الصورة التي تستخدم أثناء عرض النموذج كرسوم متحركة للتعبير عن كيانات كل نوع.

- تستخدم وحدة (Process) لتعريف وحدات معالجة الكيانات كآلة المصنع أو الصراف بالبنك، بما في ذلك المورد الذي يقوم بالمعالجة أو يوفر الخدمة، وما يلحق به من صف انتظار، وما يصاحب ذلك من تأخر زمني. وتسمح وحدة (Process) بتعريف تفاصيل مثل اسم ونوع المورد الذي سيوفر الخدمة ونوع التأخر الزمني المصاحب له من حيث التوزيع الإحصائي ومؤثراته وما إذا كان صف الانتظار للحصول على هذا المورد يطبق نظام أولوية أم لا. وكما هو الحال مع سائر وحدات أرينا، فيمكن إدخال هذه البيانات عن وحدة (Process) إما من خلال الحقول التي تفتح بالضغط المزدوج على أيقونة الوحدة في شاشة تصميم النموذج، أو من خلال نافذة وحدات البيانات، حيث سيظهر بالضغط على أيقونة الوحدة مرة واحدة جدول من نوع (Spreadsheet) مصاحب يحمل أيضاً اسم (Process) ويحتوي على صف مستقل لبيانات كل وحدة (Process) ترد في النموذج.

- ومن وحدات التسلسل المنطقي الأساسية في أرينا وحدة (Resource) والتي تعرف تفاصيل الموارد التي تتنافس عليها الكيانات في محطات المعالجة، مثل طاقة المورد الاستيعابية من الكيانات وما إذا كانت ثابتة أو تتغير حسب جدول زمني محدد مسبقاً، واحتمالية تعطل المورد عن العمل.

- تستخدم وحدة البيانات (Queue) لتعريف تفاصيل صفوف الانتظار المصاحبة لوحدات المعالجة، مثل نظام الترتيب في الصف (Discipline) وإذا كان هناك نظام أولوية لترتيب الكيانات الموجودة في الصف.

- تستخدم وحدة (Dispose) المنطقية لتمثيل خروج الكيانات المؤقتة خارج حدود النظام بعد انتهاء دورتها فيه.

ويلاحظ أن لوح (BASIC) يحتوي على المزيد من الوحدات الأخرى التي لا يتسع المقام لذكرها جميعاً هنا.

ومن الألواح الأخرى التي تتوفر في النسخة القياسية (Arena Standard) لوح (Advanced Process) الذي يحتوي على وحدات تسلسل منطقي وبيانات أكثر تعمقاً وتعقيداً من وحدات لوح (Basic). فعلى سبيل المثال:

- هناك وحدات التسلسل المنطقي (Delay) لتوقيت فترة تأخر أو انتظار للكيانات عند نقطة معينة في النموذج.
- تستخدم (Seize) لتخصيص وحدات مورد معين أو مجموعة موارد إلى كيان معين.
- تستخدم (Release) لإخلاء وحدات مورد معين سبق تخصيصه لكيان ما.
- تستخدم (Search) للبحث عن كيان معين في صف أو مجموعة في النموذج يحقق شرطاً معيناً.
- تستخدم (Remove) لإزاحة كيان من موضع معين في صف في النموذج وإرساله إلى وحدة تسلسل منطقي أخرى محددة.

كما يحتوي لوح (Advanced Process) على وحدات بيانات عدة مثل:

- تستخدم (Expression) لتعريف تعبير رياضي معين وتخصيص اسم له.
- تستخدم وحدة (Failure) لتعريف الكيفية المفترضة لتعطل موارد النموذج.
- تستخدم وحدة (Statistic) لتعريف المزيد من الإحصاءات التي يستطيع البرنامج جمعها أثناء التشغيل عن خصائص النموذج.

أما اللوح الثالث الرئيس في أرينا فهو لوح (Advanced Transfer) فيوفر وحدات تعالج تفاصيل عملية نقل الكيانات من نقطة إلى نقطة في النموذج عندما يكون هناك في النظام الأصلي ناقل يحدد بخواصه وطبيعته عدد الكيانات التي يمكن نقلها أو زمن نقلها أو كيفية نقلها. وفي هذا الصدد تعرف أرينا نوعين من النواقل: الناقل (Transporter) ويمثل العربات والروافع والسيارات والبشر، والحوامل (Conveyors) والتي تحدد القدرة على النقل بها المساحة المتوافرة عليها، مثل السلالم الكهربائية وأحزمة نقل الأمتعة بالمطارات.

ومن وحدات التسلسل المنطقي التي تتوفر في لوح (Advanced Transfer) :

- وحدة (Station) وتستخدم لتعريف محطة للوصول يتم عندها معالجة الكيانات المنقولة.

- وحدة (Enter) وتستخدم المشابهة التي تعرف - بالإضافة لتعريف المحطة - زمناً لتفريغ الكيان المنقول من الناقل وتقوم بإخلاء الناقل بعدها وإعادته إلى الاستخدام العام.
- وهناك أيضاً وحدة (Leave) التي تستخدم لنقل كيان إلى محطة حيث ينتظر عند دخولها الكيان حتى يتوفر له ناقل، ثم يمر بتأخر زمني يعبر عن زمن تحميله إلى الناقل، ثم يتم بعدها نقل الكيان إلى وجهته.

ومن الوحدات المتعلقة بنواقل (Conveyor):

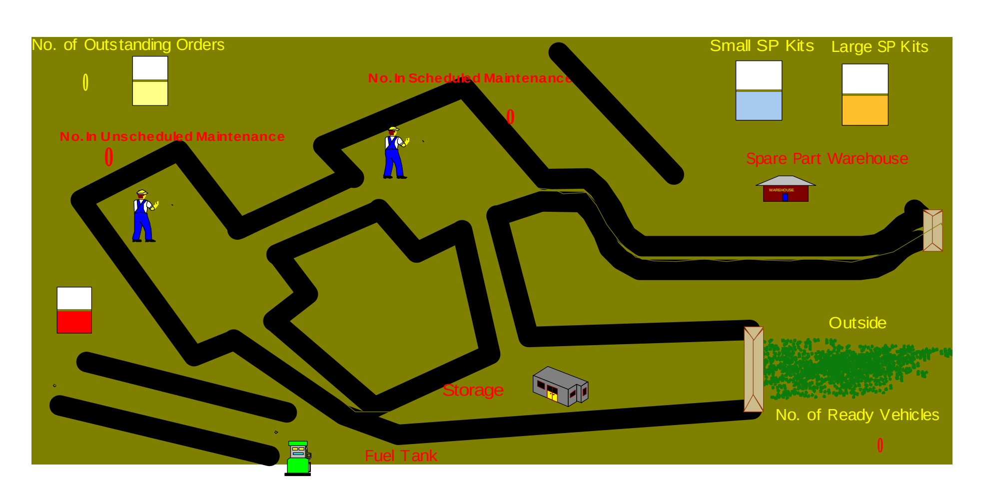
- وحدة (Access) لتخصيص واحد أو أكثر من خلايا الحامل (Conreyor) إلى كيان لنقله من محطة إلى محطة.
- وحدة (Convey) لتحريك الكيان على الحامل من محطته الحالية إلى محطة وجهته المرسل إليها.

ومن الوحدات المتعلقة بنواقل (Transporter):

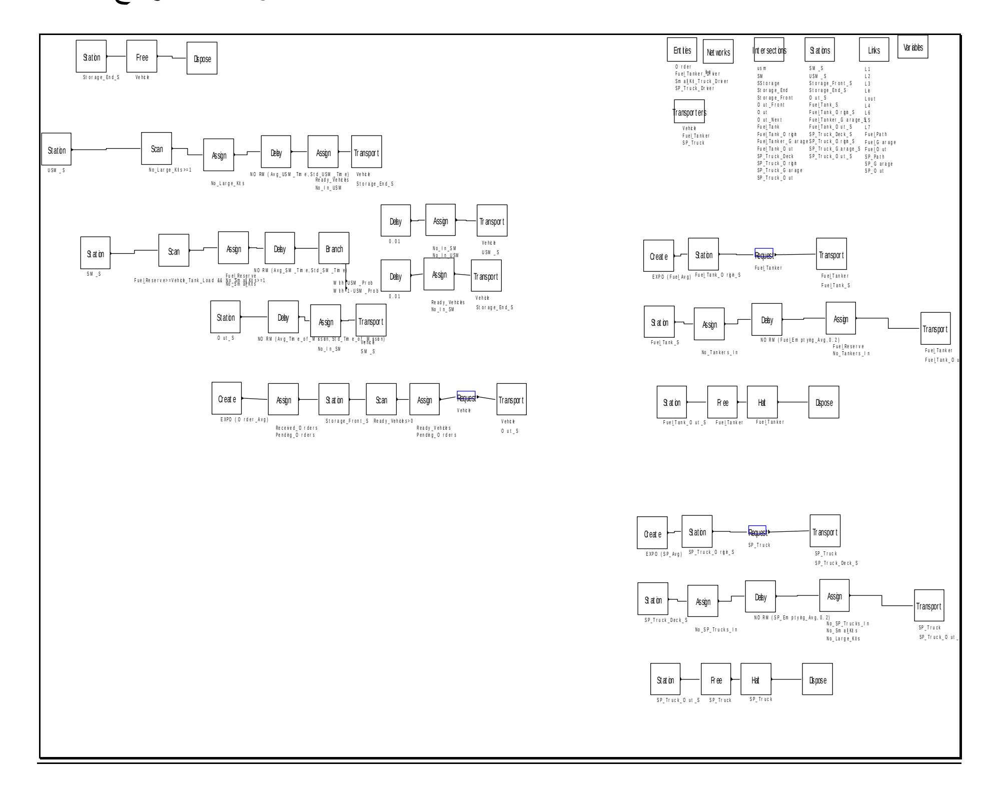
- وحدة (Allocate) لتخصيص ناقل (Transporter) إلى كيان لكن بدون تحريك الناقل إلى المحطة التي يتواجد فيها الكيان حاليًا.
- وحدة (Request) لتخصيص الناقل للكيان مع تحريكه إلى محطته.
- وحدة (Transport) لنقل الناقل (Transporter) وأيضاً الكيان الذي يحمله من محطة المصدر إلى محطة الوجهة.

وبصفة عامة توفر أرينا عدة خصائص مفيدة لمبرمجي نماذج المحاكاة مثل تحريك الرسوم (Animation) والتي تتم بصورة تلقائية بمجرد تعريف الكيانات والوحدات التي تمر بها وبمجرد اختيار الصور التي تعبر عن كل نوع من هذه الكيانات أو الموارد أو الخلفية (مثل الصور الثابتة كالأشجار أو علامات الطرق). وهناك أيضاً إمكانية تعريف النماذج التحتية (Submodels) من خلال تعريف النموذج كأجزاء هيكلية يسمى كل مستوى منها نموذجًا تحتيًا ويعرض على شاشة مستقلة، وذلك لتمكين المحلل من بناء نماذج كبيرة الحجم لا تتسع الشاشة الواحدة لعرضها.

وبالإضافة لذلك فهناك جمع الإحصاءات عن النموذج أثناء تشغيل البرنامج تلقائياً مع عرض منحنيات ومؤشرات آنية أثناء التشغيل لكيفية تطور قيمها، وعرض ملخصات وتحليلات إحصائية بعد انتهاء التشغيل عن متغيرات مخرجات النموذج. كما تتمتع أرينا بمجموعة كبيرة من التوزيعات الإحصائية المتاحة لاستخدام المحلل في تمثيل مدخلات النموذج، بالإضافة إلى إمكانية استخدام أرينا لنمذجة النظم المتصلة (Continuous Systems) وليس فقط النظم المتقطعة الأحداث (Discrete-Event Systems). ويوضح الشكل رقم (٤-١) مثالاً لشاشة عرض أحد نماذج المحاكاة المتحركة المكتوبة بحزمة أرينا ، كما يوضح الشكل رقم (٤-٢) تصميم النموذج نفسه باستخدام وحدات أرينا.



الشكل رقم (٤-١): شاشة التمثيل الحركي لأحد نماذج المحاكاة باستخدام حزمة أرينا (Arena)



الشكل رقم (٤-٢): شاشة تصميم نموذج المحاكاة نفسه باستخدام حزمة أرينا (Arena)

مثال:

لو أردنا بناء نموذج محاكاة باستخدام أرينا لتمثيل عملية تسجيل الطلاب للمقررات الجامعية في بداية الفصل الدراسي فسنجد أنها تنطوي على الخطوات التالية:

١- وصول الطالب إلى نظام التسجيل للبدء في خطواته.

٢- تلقي الإرشاد الأكاديمي من المرشد الأكاديمي للقسم.

٣- التوجه إلى مدخل البيانات لتنفيذ العمليات التي يحتويها نموذج التسجيل الموقع من المرشد الأكاديمي.

٤- إذا كانت العمليات المطلوبة كلها متاحة على نظام معلومات التسجيل على الحاسب الآلي، فإنها تنفذ ويتم طباعة الجدول الدراسي للطالب. وإذا كان بعض العمليات أو كلها غير متاح، فإن الطلب يعاد من مدخل البيانات إلى المرشد الأكاديمي للتعديل.

وعلينا الآن أن نبدأ باختيار الوحدات المنطقية التي ستمثل الخطوات التسلسلية (العملية) التي تمر بها طلبات التسجيل، والتي ستمثل الكيان الرئيس في النموذج. فلتمثيل الخطوة الأولى (وصول الطلبات- Request Arrival) سوف نختار الوحدة (Create) ونعيد تسميتها لتصبح (Request Arrival) ثم سنمثل الخطوة الثانية (تلقي الإرشاد الأكاديمي- Receive Acadmic Advising) بإستخدام وحدة (Process) مع إعادة تسميتها بالاسم المبين أعلاه. ثم لتمثيل اختيار الطلبات من حيث القابلية للتنفيذ من قبل مدخل البيانات - وبالتالي تحديد مصير الطلب إذا ما كان سينتهي بالتنفيذ وطباعة الجدول الدراسي الجديد أو إلى الإعادة لنقطة البداية مرة أخرى - فإنه بإمكاننا استخدام وحدة (Decide) من لوح (Basic) والتي تعمل بنفس عمل عبارة (If) المعروفة في لغات البرمجة، حيث تمكننا هذه العبارة من تمثيل عملية اختبار نماذج التسجيل بواسطة مدخل البيانات، والتي ستتم في هذا المثال بواسطة تحديد النسبة المئوية للنماذج القابلة للتنفيذ وللنماذج التي ستعاد لنقطة البداية مرة أخرى. وأخيراً لتمثيل نهاية مسار نماذج التسجيل - إما بالتنفيذ أو الإعادة- يمكننا إستخدام وحدة (Dispose) مع إعادة تسميتها

مرة باسم (Executed) لتمثيل خروج النماذج المنفذة من النظام، مرة باسم (Returned) لتمثيل خروج النماذج المعادة للمراجعة.

٤-٤ معايير اختيار برمجيات تطوير برامج المحاكاة :

يوجد الكثير من الصفات والقضايا التي ينبغي الالتفات إليها عند اختيار برمجيات تطوير وبناء برامج المحاكاة. وتشمل هذه الصفات والقضايا ما يلي:-

١- مدى التطابق والملاءمة بين أسلوب بناء نماذج المحاكاة في حزمة البرمجيات المرشحة وبين النظام المطلوب محاكاته.

٢- سهولة وسرعة تعلم كيفية استخدام حزمة البرمجيات.

٣- مدى سهولة أو صعوبة فهم واستيعاب نماذج المحاكاة التي يمكن تطويرها باستخدام حزمة البرمجيات من قبل المديرين وخبراء النظام من غير المتخصصين في علوم الحاسب.

٤- مقدار الدعم الفني المتوفر من الشركة المصنعة لحزمة البرمجيات.

٥- التكلفة المادية لشراء حزمة البرمجيات.

٦- سرعة تنفيذ برامج المحاكاة المطورة باستخدام حزمة البرمجيات.

٧- إمكانية عمل حزمة البرمجيات على الكثير من منصات العمل من عتاد ونظم تشغيل، أو ما يعرف في علوم الحاسب بخاصية القابلية للنقل (Portability).

٨- إمكانية دمج مكونات برمجية مكتوبة مسبقاً بإحدى لغات البرمجة مع برامج المحاكاة المطورة بإستخدام حزمة البرمجيات.

الباب الخامس

إعداد بيانات المدخلات

٥-١ المقدمة :

من العوامل المهمة التي تساعد على الحصول على نماذج محاكاة دقيقة (وبالتالي نتائج صحيحة) أن تكون بيانات المدخلات التي يغذى بها نموذج المحاكاة ذات توزيعات إحصائية مناسبة تعبر عن خصائصها الحقيقية بدقة. ومن أمثل هذه المدخلات : أزمنة الوصول وأزمنة الخدمة أو أي كميات من واقع النموذج ذات خصائص عشوائية وتم تمثيلها في النموذج كمتغيرات عشوائية وتعتبر بيانات مدخلات أي أنها خارج نطاق التحكم وليست بيانات بمخرجات. ومثل هذه الكميات تعبر عن مصادر العشوائية في النموذج وتحتاج إلى نمذجة دقيقة لخصائصها العشوائية بواسطة إحدى التوزيعات الإحصائية وبناء على ما يتوفر من عينات البيانات من هذه الكميات مما يتم قياسه ميدانياً في النظام الذي يمثله النموذج. وتتم هذه النمذجة من خلال تحليل البيانات الميدانية التي تم جمعها بواسطة بناء مخططات توزيع التكرار (Histograms) لهذه البيانات مما يمكن من تحديد التوزيعات الإحصائية المعروفة التي تكون الأقرب من حيث الخصائص وبالتالي الأدق لتمثيل التوزيعات التكرارية للبيانات.

ويعرض الملحق (أ) من هذا الكتاب عدداً من التوزيعات الإحصائية الشهيرة التي عادة ما تستخدم في تمثيل التوزيعات التكرارية لبيانات المدخلات في كثير من نماذج المحاكاة. ثم يأتي بعد ذلك خطوة تحديد قيم المؤثرات (Parameters) الخاصة بالتوزيع الإحصائي الذي وقع عليه الاختيار لتمثيل البيانات، والتي تحدد بدورها خصائص التوزيع الإحصائي،

وذلك اعتماداً على تحليل ما يتوفر من بيانات ميدانية. وبعد هذه الخطوة تأتي عملية اختبار وتقييم جودة التوزيع الإحصائي (ومؤثراته) ومدى صلاحيته لتمثيل الخصائص الإحصائية للكميات العشوائية في النموذج المراد تمثيلها. وتتم هذه العملية الاختبارية بصورة تكرارية عن طريق تطبيق أحد الاختبارات الإحصائية المعروفة على البيانات المتوافرة والتوزيع الإحصائي المختار، ثم بناء على نتيجة الاختبار، إما أن يقبل المحلل بالتوزيع ويعتمده كتمثيل صالح للمدخلات أو أن يعود ليختار توزيعاً إحصائياً آخر (بما في ذلك التوزيعات التجريبية) أو فقط ليعدل قيم مؤثرات التوزيع ، ومن ثم إعادة الاختبار مرة أخرى حتى يجتاز التوزيع الإحصائي الأخير الاختبار بنجاح. وتسمى الاختبارات الإحصائية المستخدمة هنا اختبارات جودة التطابق (Goodness-of-fit tests) ، ومن أمثلة هذه الاختبارات اختبار (Chi-square) واختبار (Kolmogorov-Smirnov) اللذان سيتم التعرف عليهما لاحقاً في هذا الفصل.

٥-٢ عملية جمع البيانات الميدانية :

تحتوي كثير من الأنظمة الواقعية على كميات ومتغيرات عشوائية ذات تأثير مهم في خصائصها ومخرجاتها . وكثيراً ما تكون هذه الكميات العشوائية كميات قابلة للقياس والمراقبة الميدانية من خلال مراقبة النظام أثناء عمله على الطبيعة (إذا كان هذا النظام قائماً بالفعل) وأخذ القراءات – إما يدوياً أو من خلال أجهزة خاصة كالعدادات الإلكترونية وأجهزة التحسس

(Sensors) – لعينات من هذه المتغيرات العشوائية. وتخضع عملية جمع البيانات في كثير من الأحيان للكثير من المحددات. فعلى سبيل المثال :

١- قد تختلف الخصائص الإحصائية للبيانات مع زمن جمع البيانات وما إذا كان ليلاً أو نهاراً أو في أيام معينة من الأسبوع، مما يستدعى أخذ عينات من البيانات في جميع هذه الأوقات بعد مراقبتها في البداية لفترة كافية لحصر الأوقات المناسبة لأخذ العينات وتحديد وسائل وأماكن جمعها وحجم البيانات اللازم جمعها لإعطاء انطباع كاف عن خصائص الكميات العشوائية .

٢- قد لا يتمكن جامع البيانات من تسجيل البيانات خلال فترات مهمة من الزمن نظراً لظروف تتعلق بالنظام المراقب أو لأسباب اجتماعية لا تتيح مراقبة الكميات العشوائية المطلوب جمع عينات من البيانات عنها. وفي هذه الحالات تختل مصداقية البيانات المجموعة ميدانياً ولا تصبح كافية للحصول على تمثيل إحصائي دقيق للكميات العشوائية موضع الدراسة.

٣- في بعض الأنظمة ، من الممكن أن يكون هناك علاقة إحصائية بينية واعتمادية بين اثنين أو أكثر من كميات المدخلات العشوائية أو الاعتماد الداخلي الذاتي لأحد الكميات (Autocorrelation) مما يتطلب استخدام وسائل رياضية أكثر تعقيداً لنمذجة مثل هذا النوع من البيانات.

٥-٣ عملية تحليل بيانات المدخلات :

بعد الحصول على بيانات المدخلات الميدانية تبدأ عملية تحليل هذه البيانات بهدف تحديد أقرب التوزيعات الإحصائية لخصائصها التي يتم التعرف إليها من خلال عملية التحليل. وتعتبر أهم وسيلة لاستخراج الخصائص الإحصائية للبيانات الميدانية والتعرف إليها هي مخططات التوزيع التكراري (Histograms). وتبنى هذه المخططات من خلال تقسيم نطاق البيانات إلى فترات متساوية أو مختلفة الأطوال ثم تحديد تكرارية البيانات المرصودة ميدانياً والواقعة في نطاق كل فترة من هذه الفترات، ومن ثم رسم قيمة التكرارية الخاصة بكل فترة رأسياً كعمود فوق هذه الفترة، بينما ترسم الفترات على المحور الأفقي. وتأتي أهمية مخططات التوزيع التكراري نظراً لأنه يمكن أن يستدل بها على دالة التوزيع الإحصائي (Probability Density Function) للمتغيرات العشوائية التي تمثلها بيانات المدخلات التي تم جمعها ميدانياً. فالنسبة للبيانات المتصلة، لو تم توصيل نقاط المركز لكل عمود من أعمدة مخطط التوزيع التكراري النسبي (Relative Frequency Distribution) والذي يكون ارتفاع كل عمود فيه يساوي تكرارية كل فترة مقسومة على العدد الكلي لعينة البيانات المشاهدة ميدانياً، ويكون مجموع المساحات الكلية المحصورة تحت جميع أعمدة الفترات فيه يساوي واحدًا. وبالمقابل، يمكن بناء مخططات توزيع تكراري للبيانات المتقطعة (discrete data) بأسلوب مشابه .

وبعد ذلك ، على المحلل الانتقال من مخطط التوزيع التكراري إلى تحديد نوعية دالة التوزيع الإحصائي الأنسب التي تمثل بدقة خصائص

البيانات الميدانية كما يظهرها مخطط التوزيع التكراري. وعادة ما يتم تحديد نوعية التوزيع من خلال تحديد فئة أو عائلة من التوزيعات التي تقارب في شكلها شكل مخطط التوزيع التكراري للبيانات. ويستخدم في هذه العملية عدد من التوزيعات الإحصائية البسيطة الشهيرة مثل (Normal) و (Exponential) وتلك الأكثر تعقيداً مثل (Gamma) و (Weibull) لتمثيل الخصائص الإحصائية للبيانات كما تظهر من شكل مخطط التوزيع التكراري (انظر الملحق (أ)). وتعتمد هذه العملية على خبرة ودراية المحلل بهذه التوزيعات الإحصائية ومعرفته أيضاً بطبيعة البيانات الميدانية وطبيعة النظام الذي جمعت منه. وقد يلجأ المحلل في بعض الأحيان إلى استخدام ما يعرف بالتوزيع التجريبي (Empirical Distribution) والذي يعتمد على البيانات الفعلية المجموعة ويستخدم عندما لا يوجد توزيع إحصائي معروف يشابه في خصائصه خصائص هذه البيانات.

وبعد أن يقوم المحلل بتحديد نوعية التوزيع الإحصائي لعينة البيانات التي تجمعها، تأتي مهمة تقدير المؤثرات الخاصة بهذا التوزيع ، ومن أمثلتها متوسط العينة (Sample Mean) وتباين العينة (Sample Variance). ولحساب هذه المؤثرات، علينا أن نستذكر المعادلات التالية. فإذا كان حجم العينة هو (n) وكانت قيم المدخل (X) المشاهدة ميدانياً هي $X_1, X_2, ...X_n$ ، فإن متوسط العينة يعرف على أنه :

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

وتباين العينة يعرف على أنه :

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n} X_i^2 - n\overline{X}^2$$

وبالنظر في تعريفات التوزيعات الإحصائية الشائعة، فسنجد أن هناك مؤشرات خاصة بكل منها قد لا تتكرر لدى التوزيعات الأخرى (نظر الملحق (أ) لمزيد من المعلومات عن التوزيعات الإحصائية). فعلى سبيل المثال، يعتمد توزيع (Normal) على مؤشرين رئيسين هما المتوسط (m) والتبيان (s^2) اللذين يمكن تقديرهما باستخدام المعادلتين المبينتين أعلاه ولكن إذا نظرنا إلى توزيع (Exponential)، فسنجد أنه يعتمد على المؤشر (l) الذي يمكن حساب تقدير لقيمته $(\hat{l})$ باستخدام المعادلة : $\hat{l} = \frac{1}{\overline{X}}$ ، حيث إن $(\overline{X})$ هو بحسب المعادلة المبينة أعلاه.

وعليه، فإنه يجب تحديد المؤشرات المعروفة لهوية نوعية التوزيعات الإحصائية التي تم تحديدها ثم حساب قيم تقديرية لهذه المؤشرات من واقع البيانات الفعلية الميدانية (وباستخدام قيم متوسط العينة وتباين العينة المحسوبتين بالمعادلتين أعلاه من واقع هذه البيانات الميدانية)، مما يمكننا من تحديد التوزيع الإحصائي النهائي الذي نعتقد أنه يمثل الخصائص الإحصائية لبيانات المدخلات الفعلية التي جمعناها ميدانياً.

٥-٤ اختبارات التأكد من جودة التطابق **(Goodness-of-fit Tests)** :

سندرس في هذا الفصل اختبارين رئيسيين للتأكد من جودة التطابق بين خصائص البيانات الفعلية المقاسة وبين خصائص التوزيع الإحصائي النهائي الذي تم التوصل إليه باتباع المنهجية المبينة في الفصل السابق ٥-٣. وهذان الاختباران هما: اختبار (Chi-Square) واختبار (Kolmogorov-Smirnov). وتنتمي هذه النوعية من الاختبارات إلى اختبارات الفرضيات الإحصائية المعروفة. وتعتمد فكرتها على وضع الفرضية التالية كفرضية باطلة (Null Hypothesis) :

ف : "إن جميع المشاهدات $(X_1, X_2,..., X_n)$ للبيان المدخل (X) هي متغيرات عشوائية مستقلة ومتماثلة التوزيع لها جميعاً التوزيع الإحصائي المعين F".

وعلى المحلل أن يسعى عندما يستخدم هذه الاختبارات إلى رفض هذه الفرضية أو قبولها باستخدام البيانات الفعلية المتوفرة. وعادة ما تفشل هذه الاختبارات في حال ما إذا كان حجم العينة صغيراً أو حتى متوسطاً لعجزها عن اكتشاف الفروق البسيطة التي قد تظهر في حجم عينة صغير بين البيانات الفعلية والتوزيع الإحصائي، حيث إنها في هذه الحالات عادة ما تقبل الفرضية الباطلة (ف) ربما لجميع التوزيعات الإحصائية المرشحة. أما في حالة النقيض الآخر عندما يكون حجم العينة كبيراً جداً فإن هذه الاختبارات تكون حساسة جداً لأي فرق بين البيانات الفعلية والتوزيع الإحصائي ، حتى أنها قد تؤدي إلى رفض الفرضية الباطلة لربما جميع التوزيعات الإحصائية المرشحة. لذا، يجب على المحلل فهم طبيعة هذه

الاختبارات جيداً واستخدامها بحذر بمفهوم أنها تعطي مؤشراً تقريبيًا استرشادياً لمدى ملاءمة التوزيع الإحصائي المختار.

٥-٤-١ اختبار **(Chi-Sqnare)** :

يقارن هذا الاختبار بين مخطط التوزيع التكراري للبيانات وبين شكل التوزيع الإحصائي المرشح، وخاصة في حالة عينات البيانات الكبيرة سواء للبيانات المتقطعة أو المتصلة. ولحساب إحصائي الاختبار، يجب علينا تقسيم كامل نطاق التوزيع الإحصائي المرشح إلى عدد (k) من الفترات المتجاورة ثم رصد عدد القراءات (X_i) التي تقع قيمها بداخل كل فترة (T_j) ، حيث أن $j = 1,2,...,K$ وحيث عدد القراءات (X_i) بداخل (T_j) هو (N_j) .

بعد ذلك نقوم بحساب النسبة المتوقعة (P_j) من القراءات التي ستقع قيمها بداخل الفترة (T_j) إلى مجموع عدد القراءات في حال ما إذا كانت الفرضية (ف) صحيحة. وتحسب هذه النسبة (في حالة البيانات المتقطعة) وفق المعادلة التالية (Law & Kelton, 2000):

$$P_j = \sum_{x_i \in T_j} \hat{P}(x_i)$$

ووفق المعادلة التالية (في حالة البيانات المتصلة) :

$$P_j = \int_{x_i \in T_j} \hat{f}(x)dx$$

وحيث $(\hat{P})$ هي دالة التوزيع الإحصائي المرشحة في حالة البيانات المتقطعة، و $(\hat{f})$ هي دالة التوزيع الإحصائي المرشحة في حالة البيانات المتصلة.

ويعرف الاختبار إحصائي الاختبار (Test Statistic) على أنه المتغير العشوائي (c^2) حيث:

$$c^2 = \sum_{j=1}^{k} \frac{(N_j - np_j)^2}{np_j}$$

حيث (n) هو حجم العينة، و (k) هو عدد الفترات، وحيث (np_j) هي القيمة المتوقعة للتكرارية في الفترة رقم (j) وفقاً للتوزيع الإحصائي المرشح، بينما (N_j) هي التكرارية الفعلية من واقع البيانات الميدانية. لذا، فإنه كلما قلت قيمة الإحصائي (c^2) كان التوزيع الإحصائي أنسب لتمثيل البيانات . لذا، فإننا سنقوم برفض الفرضية الباطلة (ف.) عندما يكون (c^2) كبيراً (أكبر من حد معين).

ويعتمد الاختبار على الحقيقة الرياضية أن المتغير العشوائي (c^2) يتبع تقريباً توزيعاً إحصائياً من نوع (Chi-Square) له عدد من درجات الحرية (degrees of freedom) يساوي : $(K-S-1)$ حيث (K) هو عدد الفترات

و (S) هـو عـدد مؤشـرات الـتوزيع الإحـصائي المرشـح موضـع الاختـبار. ويحدد الاختبار القيمة الحرجة $X^2_{a,K-S-1}$ علـى أنهـا الحد المعين الـذي بنـاء عليه تقبل الفرضـية الباطلة (ف) أو ترفض، حيث إن الفرضـية (ف) يجب أن ترفض في حال ما إذا كان :

$$c^2 > c^2_{a,K-S-1}$$

ويمكـن إيجـاد القـيمة $c^2_{a,K-S-1}$ مـن الجـداول الإحـصائية لـتوزيع (Chi-Square) والمعروفة بكتب الإحصاء مثل (Walpole & Myers, 1985) ، حيث (a) هـي درجـة الأهمـية المطلـوبة (degree of significance) وتـسمى الـنقطة $c^2_{a,K-S-1}$ النقطـة الحـرجة لتوزيع (Chi-Square) ، علمـاً بـأن الاختـبار المبين أعلاه لا يصح إلا في الحالات التي يكون حجم العينة (n) فيها كبيراً.

مثال :

إذا أعطـيت مجموعـة مـن البيانـات التـي تمثل عدد الزبائن الذين يدخلون فرع أحد البنوك في فترة النصف ساعة الأولى من ساعات الدوام الصباحية .

استخدم اختبار (Chi-Square) لتحديد أقـرب الـتوزيعات الإحـصائية الـتي تمثل هذه العينـة من البيانات.

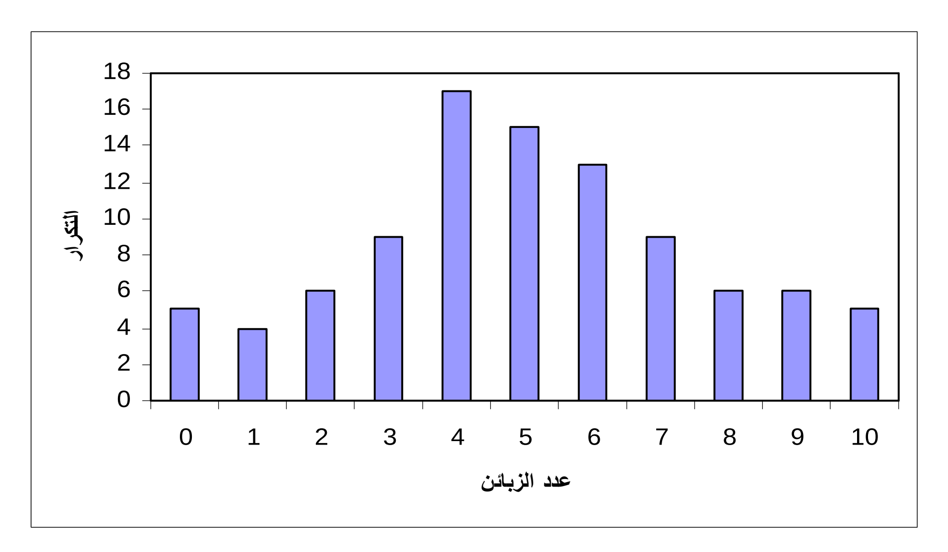
الحل:

لنفرض أن عينة البيانات الميدانية التي تم قياسها عن المتغير العشوائي "عدد الزبائن الداخلين خلال النصف ساعة الأولى من دوام فرع البنك"، قد تم تحويلها لتصبح بالصورة المبينة في الجدول التالي:

الجدول رقم (٥-١) : عدد الزبائن الواصلين لفرع البنك في أول نصف ساعة

عدد الزبائن الواصلين	التكرار (عدد الأيام)
٠	٥
١	٤
٢	٦
٣	٩
٤	١٧
٥	١٥
٦	١٣
٧	٩
٨	٦
٩	٦
١٠	٥

ونلاحظ من جدول (٥-١) أن عدد أيام مراقبة وصول الزبائن إلى البنك يبلغ ٩٠ يوماً. ولتحديد توزيع إحصائي تقريبي لهذه البيانات، علينا أن نبدأ برسم مخطط التوزيع الإحصائي (Histogram) والمبين في الشكل رقم (٥-١).



الشكل رقم (٥-١) : مخطط التوزيع الإحصائي للبيانات

بعد ذلك، يجب علينا ترشيح أحد التوزيعات الإحصائية المتقطعة – وفقاً لشكل المخطط الوارد في الشكل رقم (٥-١) – ليكون تمثيلاً لهذه البيانات. وبناء عليه، فإننا نرشح توزيع بويسون (Poisson) . وبحساب المتوسط لعدد الزبائن الواصلين في الفترة المحددة نجد أنهم ٥٫٠٤٢ أي أن إحصائي توزيع بويسون التقديري (a) هو ٥٫٠٤٢ . بعد ذلك، يجب علينا وضع الفرضيات المتعلقة باختبار (Chi-Square) وهي كالتالي (راجع الباب السادس – فقرة ٦-٤):

ف. : "إن المتغير العشوائي لوصول الزبائن خلال المدة المحددة موزع بحسب توزيع بويسون".

ف١ : "إن المتغير العشوائي لوصول الزبائن خلال المدة المحددة غير موزع بحسب توزيع بويسون".

وكما نعرف (راجع الملحق أ)، فإن توزيع بويسون هو توزيع إحصائي متقطع له دالة كثافة إحصائية لها الصيغة التالية:

$$P(x) = \begin{cases} \frac{e^{-a} a^{x}}{x!} & x = 0,1,2,... \\ 0 & Otherwise \end{cases}$$

وبالتعويض في الصيغة أعلاه بالقيمة ٥ر٠٤٢ = a ، فإنه بإمكاننا الحصول على القيم P_j في معادلة حساب المتغير العشوائي c^2 الخاص بالاختبار. وبعد ضرب كل P_j في حجم العينة n هو ٩٠، فانه بإمكاننا الحصول على قيم np_j بحسب الجدول رقم (٥-٢).

الجدول رقم (٥-٢): الحصول على قيم np_j

np_j	j

٠	٠ر٦١٣
١	٣ر٠٩٥
٢	٧ر٨٠٢
٣	١٣ر١١٢
٤	١٦ر٥٢٨
٥	١٦ر٦٦٦
٦	١٤ر٠٠٥
٧	١٠ر٠٨٨
٨	٦ر٣٥٨
٩	٣ر٥٦٢
١٠	١ر٧٩٦

أما عن قيم N_j في صيغة حساب (c^2) ، فهي تمثل عدد مرات التكرار في المخطط في الشكل رقم (٥-١) والتي يعبر عنها ارتفاع كل عمود في المخطط. وبالتالي، يمكننا الآن حساب قيمة (c^2) وسنجد أنها: ١ر٤١ = c^2 .

ونظراً لأن عدد مؤشرات توزيع بويسون هو $(S=1)$ ، وبما أن عدد الفترات هو (K =١٠) ، فإن عدد درجات الحرية لتوزيع c^2 هو (٨ -K) =S-1 . وإذا افترضنا درجة أهمية بمقدار (٠٥ر٠)، فإنه بالرجوع إلى الجداول الإحصائية (Walpole & Mayers, 1985) نجد أن قيمة $c^2_{0.05,8} = 10.0$. وبما أن قيمة c^2 المحسوبة هي ٤١,١، فإن استخدام اختبار (Chi-Square) يؤدي إلى رفض الفرضية ف. والتي تفترض أن

التوزيع الإحصائي المناسب لهذه البيانات هو من نوع بويسون، أي أنه يلزم البحث عن توزيع آخر أكثر ملاءمة لهذه البيانات.

٥-٤-٢ اختبار (Kolmogorov-Smirnov):

يعتبر هذا الاختبار (ويختصر اسمه إلى اختبار K-S) من الاختبارات الصالحة للاستخدام في الحالات التي يكون فيها حجم العينة صغيراً، وهو لا يتطلب تكوين فترات من البيانات، كما أنه يعطي نتائج جيدة عند المقارنة بمجموعات أكثر عدداً من التوزيعات الإحصائية المرشحة. ويعرف إحصائي اختبار (K-S) وفقاً للصيغة (Law & Kelton, 2000) التالية :

$$D_n = \max\left\{D_n^{+}, D_n^{-}\right\}$$

حيث
$$D_n^{+} = \max_i\left\{\frac{i}{n} - \hat{F}(X_i)\right\}$$

و
$$D_n^{-} = \max_i\left\{\hat{F}(X_i) - \frac{i-1}{n}\right\}$$

وحيث $\hat{F}(x)$ هي دالة التوزيع الإحصائي التراكمي المرشحة، و X_i هو المتغير العشوائي الذي يعبر عن القراءة الميدانية رقم i لبيان مدخل معين من مجموع عينة من n قراءات ميدانية لهذا البيان. وتعتمد فكرة اختبار $(K-S)$ على مقارنة دالة التوزيع الإحصائي التراكمي (المتصلة) المرشحة لتمثيل البيانات الميدانية ودالة التوزيع الإحصائي $\hat{F}(X_i)$

التراكمي للبيانات الميدانية والتي يمكن تعريفها بالصيغة التالية لعينة من القراءات ذات حجم n:

$F_n(x)$ = (عدد القراءات X_i التي هي أقل من أو تساوي القيمة x) $(1/n)$

ويلاحظ هنا أن $F_n(X_i)$ هي دالة درجية تعطي بالصيغة:

$$F_n(X_i) = \frac{i}{n}, \qquad i = 1,2,\ldots,n$$

ومن ثم فإن القيمة D_n^+ تمثل أكبر فرق بين قيمة كل قراءة $\hat{F}(X_i)$ من الدالة المرشحة $\hat{F}$ وبين قيمة الدالة الدرجية للبيانات الميدانية $F(X_i)$ التي تعلوها مباشرة، و D_n^- تمثل أكبر فرق بين قيمة كل قراءة $\hat{F}(X_i)$ من الدالة المرشحة $\hat{F}$ وبين قيمة الدالة الدرجية للبيانات الميدانية $F(X_i)$ التي تقع تحتها مباشرة.

وبالتالي، يصبح معنى الإحصائي D_n هو أكبر فرق علوي أو سفلي بين قيم دالتي التوزيع الإحصائي التراكمي للدالة المرشحة ودالة البيانات الميدانية عند كل قراءة ميدانية X_i لكل قيم $i = 1,2,\ldots,n$

لذا ، فإن الحصول على قيمة كبيرة للإحصائي D_n تفوق القيمة الثابتة $d_{n,1-a}$ والتي تسمى النقطة الحرجة للاختبار يؤدي إلى رفض الفرضية الباطلة والتي تفترض أن التوزيع الإحصائي المرشح يستطيع تمثيل خصائص البيانات الميدانية للمدخلات، وحيث القيمة (a) هي مستوى

الاختبار والتي يحددها المحلل، وحيث تعتمد قيمة $d_{n,1-a}$ على نوعية التوزيع الإحصائي المرشح. وبالرجوع إلى كتب الإحصاء (Law & Kelton, 2000) ، فإن صيغة اختبار $(K-S)$ يمكن إعادة صياغتها لتصبح كالتالي :

$$f(n)D_n > C_{1-a}$$

حيث $f(n)$ هي صيغة تعتمد على نوعية التوزيع الإحصائي المرشح و C_{1-a} هي قيم حرجة يمكن الكشف عنها في الجداول الإحصائية لكل قيمة من $(1-a)$ يطلبها المحلل. فعلى سبيل المثال، للتوزيع الإحصائي المرشح (Normal) والذي تم تقدير قيمة المتوسط له بالقيمة المحسوبة $\bar{X}(n)$ وقيمة التباين له بالقيمة المحسوبة $S^2(n)$ ، وحيث n هو حجم العينة، فإن اختبار (K-S) يصبح اختبار ما إذا كان:

$$(\sqrt{n} - 0.01 + \frac{0.85}{\sqrt{n}})D_n > C_{1-a}$$

وللتوزيع الإحصائي المرشح (Exponential) والذي تم تقدير قيمة المتوسط له بالقيمة المحسوبة $\bar{X}(n)$ ، فإن اختبار (K-S) يصبح اختبار ما إذا كان:

$$(D_n - \frac{0.2}{n})(\sqrt{n} + 0.26 + \frac{0.5}{\sqrt{n}}) > C_{1-a}$$

مثال :

إذا أعطينا عينة من ٥٠ قراءة لبيانات الزمن البيني لوصول الزبائن إلى فرع أحد البنوك مأخوذة على مدى فترة مراقبة من ١٠٠ دقيقة، ونريد اختبار ما إذا كان المتغير العشوائي (X) (الزمن البيني للوصول) يتبع توزيعًا إحصائيًا من نوع (Exponential)، فعلينا أن نحسب قيمة D_{50}^{+} وقيمة D_{50}^{-} باستخدام الدالة :

$$\hat{F}(x) = 1 - e^{-x/\hat{a}}$$

حيث $\hat{a}$ هي القيمة المحسوبة لتقدير متوسط الزمن البينى (X)، ومن ثم نقوم بحساب الإحصائي D_n . فإذا افترضنا أن القيمة المحسوبة لـ D_{50} هي ٠٣٩ر٠، فبإمكاننا التعويض في الصيغة المعدلة لاختبار (K-S) لتوزيع (Exponential) كالتالي :

$$(D_{50} - \frac{0.2}{50}) \ (\sqrt{50} \ + 0.26 + \frac{0.5}{\sqrt{50}}) > C_{1-a}$$

وإذا حددنا $a = .,1$ فإن $(1-a)$ تكون ٩ر٠

وبالرجوع إلى الجدول الإحصائي المناسب (Law & Kelton, 2000) نجد أن $C_{0.9}$ في هذا الحالة هي ٩٩ر٠ . وبالتعويض في الصيغة أعلاه نجد أن الطرف الأيسر من الصيغة يعطينا ٢٥٩١ر٠، بينما الطرف الأيمن قيمته ٩ر٠، أي أن الفرضية الباطلة (ف.) لا ينبغي أن ترفض في هذه الحالة عند مستوى ١ر٠ = a .

الباب السادس

التحقق من صحة وصلاحية النموذج

٦-١ المقدمة :

ذكرنا فيما مضى أن من مراحل دراسات المحاكاة المهمة مرحلة التحقيق من صحة وصلاحية النموذج. وتهدف هذه المرحلة إلى التأكد بأسلوب علمي من أن نموذج المحاكاة الذي تم تطويره يمثل النظام الفعلي بدرجة كافية لاعتباره بديلاً ـ لأغراض الدراسة واستخراج البيانات ـ عن النظام الأصلي. كما تهدف أيضاً هذه المرحلة إلى رفع درجة ثقة متخذي القرار من المديرين وزيادة ارتياحهم إلى مصداقية نموذج المحاكاة المطور.

وتنقسم مهام هذه المرحلة إلى قسمين: التحقق من صحة النموذج (Verification Model) والتحقق من صلاحية النموذج (Model Validation). وتهتم مهمة التحقق من صلاحية النموذج بالمقارنة بين النموذج التصوري الذي توصل إليه المحلل عن النظام الأصلي وبين برنامج المحاكاة الذي تم تطويره. فبالتالي، يقصد بصحة النموذج هنا صحة برنامج المحاكاة من حيث مدى تطابقه مع معطيات النظام المحددة في النموذج التصوري. أما مهمة التحقق من صلاحية النموذج فتهتم بالتأكد من أن برنامج المحاكاة الذي تم تطويره يمثل بدقة ويتطابق في السلوك والخصائص (بدرجة كافية) مع النظام الأصلي، فبالتالي، تتم هذه المهمة بمقارنة برنامج المحاكاة مع النظام الأصلي والتأكد من أن الفرق بينها أقل من حد معين يضعه المحلل كمعيار لقبول نموذج المحاكاة أو رفضه. وفي حال عدم اجتياز نموذج المحاكاة لهذا الاختبار، فعلى المحلل العودة مرة أخرى إلى النموذج التصوري لمراجعة الفرضيات والمعطيات التي بني عليها، وكذلك مراجعة كيفية ترجمة هذا النموذج إلى برنامج محاكاة. وتتكرر هذه العملية مرات عدة إلى أن يتم

الوصـول بـنموذج المحاكـاة إلـى الدقـة المطلـوبة لاجتـياز اختـبار صـلاحية النموذج.

٦-٢ التحقق من صحة نموذج المحاكاة :

من المعلوم أن بناء نموذج المحاكاة يبدأ ببناء نموذج تصوري يجمع ويلخص ويصنف جميع الفرضيات حول هيكل النظام وقيم مؤثراته (Parameters)، وذلك بعد مراقبة النظام وتحليله. بعد هذه الخطوة يقوم المحلل بمراجعة ما حصل عليه من معلومات وفرضيات حول النظام ومضاهاتها مع النموذج التصوري الذي قام بتصميمه. فإذا انتهت هذه المراجعة بقبول النموذج التصوري، يبدأ المحلل عندئذ في تصميم وبناء برنامج (نموذج) المحاكاة الذي يمثل ما ورد في النموذج التصوري من تفاصيل على شكل برنامج حاسوبي. ويأتي دور عملية التحقق من صحة النموذج لتقارن بين ما تم تضمينه في النموذج التصوري من تفاصيل وبين الكيفية التي تمت بها عملية برمجة هذه التفاصيل في برنامج المحاكاة. وتشمل هذه التفاصيل الفرضيات عن بنية النظام الأصلي ومكوناته وقيم مؤثراته، وأي تفاصيل أخرى ذات فائدة في تكوين تصور واضح وتجريد مبسط لتفاصيل النظام الأصلي.

وتعتمد الكيفية التي تتم بها عملية التحقق من صحة النموذج على أساليب بديهية تتراوح ما بين اختبار صحة خوارزميات البرنامج ومدى مطابقة ما بها من تسلسل منطقي مع معطيات النموذج التصوري من قبل المبرمج نفسه، إلى إجراء اختبارات عن طريق تحليل النتائج العددية

المرحلية الخارجة من المراحل المختلفة بالنموذج للتأكد من مدى معقوليتها وموافقتها لما يتوقعه المحللون وخبراء النظام، وذلك لمجموعة من المدخلات والظروف الافتراضية المختلفة للنظام.كذلك، من الممكن أن يستعان بخاصية تحريك الرسوم (Animation) والمتوفرة غالباً في معظم أدوات تطوير برامج المحاكاة من أجل إعطاء صورة رسومية لكيفية سير نموذج المحاكاة والخطوات التي تحدث بداخله أثناء التشغيل. وتساعد هذه الصورة الرسومية المراقبين من محللين وخبراء على فهم تمثيل النظام بداخل النموذج بشكل سريع واكتشاف أي تناقضات بينهما بمجرد المشاهدة وقبل الخوض في تحليلات عددية مطولة. كما يساعد التوثيق الجيد لبرنامج المحاكاة بكل ما فيه من تفاصيل ومتغيرات وإجراءات على تتبع هذا البرنامج واكتشاف أوجه الاختلاف عن النموذج التصوري وعلاج هذا الاختلاف.

ومن الأساليب المستخدمة كثيراً لإنجاز مرحلة التحقق من صحة النموذج استخدام النماذج الرياضية لحساب قيم بعض مقاييس أداء النظام المرحلية (أي لجزء معين يمثل مكونًا جزئيًا للنظام الأصلي، وليس بالطبع لكامل النموذج نظراً لصعوبة ذلك). ويستفاد من هذه القيم الرياضية عن طريق مقارنتها بما يخرجه نموذج المحاكاة من قيم لنفس مقاييس الأداء للمكون الجزئي من النموذج، حيث تعتبر القيم الرياضية هي القيم الصحيحة لمقاييس الأداء (إذا لم تكن مبنية على تقريبات حسابية غير دقيقة). لذا، فهذه المقارنة سوف تعطي المحلل مؤشراً واضحاً عن مدى دقة مخرجات نموذج المحاكاة بحسب مدى مطابقتها للقيم الرياضية المحسوبة.

٦- ٣ التحقق من صلاحية نموذج المحاكاة :

تنطوي عملية التحقق من صلاحية نموذج المحاكاة على مقارنة هذا النموذج مع النظام الأصلي ثم إجراء التعديلات اللازمة على النموذج في حال ما إذا كان هناك فروق مهمة ثم إجراء المقارنة مع النظام مرة أخرى، وهكذا حتى يتم التأكد من أن النموذج أصبح تمثيلاً صادقاً ودقيقاً للنظام لأغراض الدراسة الحالية. وتنقسم أساليب إجراء هذه المقارنة بحسب ما إذا كانت مبنية على اختبارات موضوعية (Objective) أو شخصية (Subjective). ويقصد هنا بالاختبارات الشخصية أنها تعتمد على خبرات ومعارف وآراء الأشخاص من خبراء ومديري النظام حول خصائص ومخرجات نموذج المحاكاة ومدى مشابهتها لخصائص ومخرجات النظام الأصلي بحسب رأيهم. أما الاختبارات الموضوعية فهي الاختبارات الكمية التي تعتمد على إخضاع بيانات مخرجات نموذج المحاكاة وبيانات مخرجات النظام الأصلي لاختبارات إحصائية معروفة لحساب وتقييم درجة التشابه والتماثل بينهما. ومن المنهجيات الشائعة لإجراء عملية التحقق من صلاحية النموذج المنهجية التالية (Banks et. al., 2001):

١- بناء نموذج محاكاة يتمتع بصلاحية ظاهرية (أي أنه يشبه ظاهرياً النظام الأصلي في الخصائص والسلوك).

٢- مراجعة وتدقيق فرضيات هذا النموذج والتأكد من مطابقتها للنظام الأصلي.

٣- مقارنة دالة تحويل المدخلات إلى مخرجات في كل من النموذج والنظام الأصلي (أي أن النموذج والنظام يحولان المدخلات إلى مخرجات بالكيفية نفسها).

وللتحقق من التشابه الظاهري لنموذج المحاكاة مع النظام الأصلي، على المحلل الاستعانة بالمديرين وخبراء النظام لتدقيقه والتأكد من مشابهته ـ على الأقل ظاهرياً ـ في الخصائص والسلوك للنظام الأصلي في الكثير من الأحوال والظروف. ولمراجعة وتدقيق فرضيات النموذج فعلى المحلل أيضاً الاستعانة بخبرات ومعارف خبراء النظام للإفادة عن مدى دقة تلك الفرضيات وكذلك التبسيط والتجريد الذي أجراه المحلل على خصائص النظام الأصلي للخروج بهذا النموذج. وتشمل هذه المراجعة في العادة كلاً من هيكل وأسلوب عمل النظام والخصائص الإحصائية والعددية لمؤثرات ومدخلات النظام. وقد فصل الباب الخامس عدداً من الأساليب الرياضية التي يمكن استخدامها لمقارنة الخصائص الإحصائية لمجموعتين من البيانات ومدى تطابقهما، مثل توزيعهما الإحصائي وقيم المؤثرات الإحصائية لكل منهما.

أما عن مقارنة دالة تحويل المدخلات إلى مخرجات في كل من النموذج والنظام الأصلي، فتتم هذه الخطوة اعتماداً على أساليب اختبارات الفرضيات الإحصائية المعروفة (Hypothesis Testing). ويعامل النموذج أثناء هذه العملية كصندوق أسود يحتوي على دالة تحويلية (Transformation Function) تقوم بتحويل المدخلات إلى مخرجات. ولاختبار مدى تطابق هذه الدالة في كل من النموذج والنظام الأصلي، على المحلل أن يختار مجموعة من

بيانات مدخلات النظام ويقوم بتجريبها على نموذج المحاكاة ثم يقارن قيم المخرجات التي يعطيها نموذج المحاكاة مع المخرجات المسجلة ميدانياً من النظام الأصلي نتيجة إدخال نفس مجموعة بيانات المدخلات المختارة له وتحت نفس الظروف التشغيلية. ثم يقوم المحلل بإجراء اختبارات الفرضيات الإحصائية على مجموعتي مخرجات النموذج ومخرجات النظام لمعرفة ما إذا كانتا متطابقتين بالقدر الكافي. فإذا وجد أن درجة التطابق بينهما كافية، فيعلن أن نموذج المحاكاة صالح لتمثيل النظام الأصلي بدقة، وإذا ما وجد غير ذلك فعلى المحلل العودة إلى النموذج التصوري ومراجعة تفاصيله وتعديلها بما يكفل تقريب نموذج المحاكاة من النظام الأصلي في السلوكات والمخرجات بشكل أكبر، ثم يعاد إجراء الاختبارات السابقة مرة أخرى. ويلاحظ هنا أنه لإجراء اختبارات الدالة التحويلية بين النموذج والنظام، فلابد من وجود النظام الأصلي على أرض الواقع وبصورة تسمح بجمع البيانات والمشاهدات الميدانية منه. ويعني ذلك أنه لا يمكن إجراء هذه الاختبارات في الأحوال التي يبنى فيها نموذج المحاكاة مبكراً لنظام خاضع للتصميم ولم ينشأ بعد.

٦-٤ اختبارات الفرضيات الإحصائية واستخدامها في التحقق من صلاحية الدالة التحويلية لنماذج المحاكاة :

تعرف الفرضية الإحصائية على أنها ادعاء أو افتراض حول مجموعة من البيانات إذا أخضعت للاختبار، فإنها إما أن ترفض (وهو ما يعني أنها خاطئة) أو أن تقبل (وهو ما يعني ضمنياً أننا لا نملك الدليل الكافي للاعتقاد بخطئها). وعادة ما تسمى الفرضية الأساسية التي نريد اختبارها (H_0)

بالفرضـية الـباطلة (Null Hypothesis)، وتـسمى الفرضـية العكـسية لهـذه الفرضـية بالفرضـية الـبديلة (Alternative Hypothesis). وتعـرف الفتـرة الحـرجة (Region Critical) علـى أنهـا النطاق من قيم متغير ما (X) - يسمى بمتغير القرار - والذي إذا وقع (X) بداخلـه نستطيع رفض (H_0). وتعرف الفترة المكملة للفترة الحرجة بأنها المنطقة التي إذا وقع (X) بها فسوف نقبل (H_0)، وتسمى لذلك بفترة القبول (Acceptance Region) ويرمز لها بـ (H_1). ويعرف النوع الأول من الأخطـاء بأنـه رفض الفرضـية الباطلة عندما تكون صحيحة، ويعرف النوع الثاني من الأخطاء على أنه قبول الفرضية الباطلة عندما تكون غير صحيحة. وعليه نعرف درجة الأهمية (Level of Significance) (a) علـى أنـه احتمالـية ارتكـاب خطـأ مـن الـنوع الأول كالتالي:

a = {احتمال خطأ من النوع الأول }

مثال: إذا افترضنا أن الفرضية الباطلة هي أن: متوسط أطوال طلاب الجامعة هو ١٧٠سم. فعليه تكون:

H_0: المتوسط = ١٧٠ سم

H_1: المتوسط ≠ ١٧٠ سم

وفي هذه الحالة فإن الفترة الحرجة قد تكون على سبيل المثال:

(∞ ،١٧١) Y (١٦٩، -∞)

وبالتالي تكون فترة القبول هي الفترة: [١٦٩،١٧١].

بالنظر إلى المثال السابق، وإذا افترضنا أن متغير القرار هو **T**: طول الطالب الواحد بالجامعة، وإذا افترضنا أن المتوسط الفعلي للإحصائي "متوسط طول الطالب" هو q_T والانحراف المعياري الفعلي هو s_T ، وإذا أخذنا عينة من طلاب الجامعة عددها n طالب، فإن درجة الأهمية a يمكن أن تحسب كالتالي:

$$a = \Pr \{ m_T > 171 \mid q_T = 170\} + \Pr \{ m_T < 169 \mid q_T \quad 170\}$$

حيث إن m_T هو متوسط العينة المحسوب للعينة المفترضة من الطلاب (أي أن m_T هو متغير عشوائي)، و q_T هو المتوسط الفعلي لهذه العينة (ثابت). ويظهر لنا مما سبق أنه لحساب قيمة a فإنه يلزم أن يكون معروفاً لدينا التوزيع الإحصائي للمتغير العشوائي m_T. وحسب نظرية النهاية المركزية (Central Limit Theorem) الإحصائية، فإنه يمكن افتراض هذا التوزيع الإحصائي على أنه تقريباً من نوع التوزيع الطبيعي (Normal Distribution) بشرط أن يكون حجم العينة كبيراً بالدرجة الكافية. وبالتحديد، فإنه كلما اقتربت n من ∞ ،فإن دالة الكثافة الإحصائية للمتغير z المعرف على أنه:

$$z = \frac{m_T - q_T}{s_T/\sqrt{n}}$$

سوف تؤول إلى دالة التوزيع الإحصائي الطبيعي القياسي (Standard Normal) والذي يُعرف بأن متوسط متغيره العشوائي هو الصفر وأن الانحراف المعياري لمتغيره هو الواحد. ويظهر مما سبق أن علاقة المتغير العشوائي (z) ذي التوزيع الطبيعي القياسي بالمتغير العشوائي (إحصائي القرار) m_T تعني أن m_T هو أيضاً متغير عشوائي طبيعي التوزيع (Normal

) لكــنه ذو متوســط q_T وانحـــراف معــياري $s_T/\sqrt{n}$، حــيث أن s_T هــو الانحراف المعياري الفعلي للمتغير T، و n هو حجم العينة. لذا، إذا افترضنا أن s_T = ٣,٦ ، n =٣٦، فإنـه لحساب قـيمة a يمكننا عمل التعويضات التالية:

$$z_1= (169-170) / (3.6/6) = -1/0.6 = -1.67 \text{ cm}$$

$$z_2= (171-170) / (3.6/6) = 1/0.6 = 1.67 \text{ cm}$$

وعليه فإن:

$$a = \Pr\{z<-1.67\} + \Pr\{z>1.67\}$$

وبالرجوع إلى جداول التوزيع الطبيعي القياسي ، فإن:

$$a = 2 \Pr\{z>1.67\} = 0.095$$

ولتخفيض احتمالية ارتكاب خطأ من النوع الأول (a) يمكننا زيادة حجم العينـة (n) لأن ذلـك يـزيد مـن دقـة الاختـبار ويـؤدي بإحـصائي القـرار (Decision Statistic) بـأن يقترب من القيمة الفعلية. وجدير بالذكر أن زيادة حجم العينة يؤدي أيضاً إلى خفض احتمالية ارتكاب أخطاء من النوع الثاني.

أمـا فـي حـال عـدم معـرفة قـيمة الانحـراف المعـياري الفعلـي للمتغيـر العشوائي m_T، وفـي حـال كـون حجـم العينـة صـغيرًا نسبياً (n<30) فإنه من الأدق استخـدام تـوزيع (t-distribution) بدلاً من التوزيع الطبيعي القياسي في تقرير قيمة a. أي أنه يفضل استخدام التعويض التالي في هذه الحالة:

$$t = \frac{m_T - q_T}{s_T/\sqrt{n}}$$

حيث إن t هو متغير عشوائي موزع إحصائياً حسب توزيع (t-distribution) وs_T هو الانحراف المعياري للمتغير (T) المحسوب من العينة ذات الحجم n .

٦-٤-١ استخدام اختبارات الفرضيات الإحصائية في التحقق من صلاحية الدالة التحويلية لنماذج المحاكاة :

إذا افترضنا أن القيمة التي تم حسابها من البيانات المسجلة ميدانيًّا لأحد مقاييس الأداء عند مراقبة النظام الأصلي هي y. وإذا افترضنا أنه قد تم بناء نموذج محاكاة لهذا النظام يقوم بحساب قيمة مقياس الأداء نفسها في كل تشغيلة لبرنامج المحاكاة - ويرمز لهذه القيمة في التشغيلة رقم r بالرمزx_r - فإنه لاختبار صلاحية هذا النظام لتقدير قيمة مقياس الأداء ذي القيمة الميدانية y باستخدام القيم:

$\{x_r: r = 1, 2, \ldots, k\}$، فإنه يمكن تعريف الفرضية الباطلة والفرضية البديلة كالتالي:

$$H_0: E\{x\} = y$$

$$H_1: E\{x\} \neq y$$

حيث $E\{\}$ هو المتوسط. فإذا تم رفض H_0 فإنه ينبغي للمحلل العودة لتحسين النموذج ثم إعادة الاختبار وإذا ما تم قبول H_0 فإن هذا يعني أنه لا

يوجد ما يبرر رفض النموذج أو الطعن في صلاحيته (وإن كان هذا لا يثبت بالدليل القاطع أنه نموذج صالح).

ونظـراً لأن الانحـراف المعـياري الفعلـي للمتغيـر العـشوائي x غيـر معروف، ونظراً لأنه لا يتوقع أن يتجاوز عدد مرات تشغيل برنامج المحاكاة ٣٠ مرة، فإننا سنستخدم توزيع (t-distribution) لإجراء اختبار الفرضية الإحصائية عن طريق إجراء التعويض التالي:

$$t = \frac{\mathrm{m_x} - y}{\mathrm{s_x}/\sqrt{\mathrm{n}}}$$

حيث إن m_x هو المتوسط المحسوب للعينة، و s_x الانحراف المعياري المحسوب للعينـة، و n هـو حجـم العيـنة. فـإذا تـم الـتعويض بهـذه القـيم فـي الصيغة السابقة ووجد أن القيمة المطلقة لـ t الناتجة أكبر من $t_{a/2,n-1}$ ، حيث أن n-1 هي درجات الحرية (Degrees of freedom) للتوزيع (t-distribution) و a هـي درجة الأهمية المطلوبة للاختبار الإحصائي، فإن هذا يعني أن H_0 ينبغي أن ترفض، وإذا كان غير ذلك فإن H_0 ينبغي أن تقبل. أي أنه:

إذا كان $|t| > t_{a/2,n-1}$

ارفض H_0

وإذا كان $|t| \leq t_{a/2,n-1}$

اقبل H_0. ويسمى الاختبار السابق اختبار (t-distribution) الثنائي الجهة.

مثال :

إذا أعطيت عينه من ٢٠ قراءة مستخرجة من ٢٠ مرة تشغيل لبرنامج محاكاة لمقياس أداء أحد الأنظمة، وكان متوسط هذه القراءات هو ١٢ وانحرافها المعياري هو ٣. وإذا افترضنا أن القيمة المدونة لهذا المقياس عند المراقبة الميدانية هي ١٠ ، فما مدى صلاحية نموذج المحاكاة من حيث دقة تمثيله للدالة التحويلية للنظام ؟ استخدم a = ٠,٠٥

الحل:

في هذا المثال y = ١٠، m_x =١٢ ، s_x = ٣ ، n =٢٠ . وبما أن n > ٣٠ والانحراف المعياري الفعلي غير معروف، فسوف نستخدم التوزيع الإحصائي (t-distribution) لإجراء الاختبار الإحصائي اللازم.

وفيما يلي تعريف الفرضية الباطلة H_0 والفرضية البديلة H_1 :

$$H_0: E\{x\}=10$$

$$H_1: E\{x\}\neq 10$$

ولاختبار الفرضية الإحصائية سوف نحسب قيمة t كالتالي:

$$t = (12-10)/(3/\sqrt{20}) = 2.98$$

وبالكشف في جداول التوزيع الإحصائي (t-distribution) نجد أن:

$$t_{0.025,19} = 2.09$$

وبما أن 2.09<2.98

فإننا نـرفض H_0 ونعتبر أن نموذج المحاكاة ليس دقيقاً بالقدر الذي نريده في تمثيله للدالة التحويلية للمدخلات إلى مخرجات في النظام الأصلي.

الباب السابع

تحليل بيانات المخرجات

٧-١ المقدمة :

في هذا الباب سوف ندرس كيفية التحليل الإحصائي لبيانات المخرجات الناجمة عن تشغيل نموذج المحاكاة العشوائي . وقد مر علينا سابقاً أن مدخلات نموذج المحاكاة للأنظمة العشوائية لا تخلو من بيانات عشوائية ، مما يجعل بيانات مخرجات المحاكاة أيضاً عشوائية بحيث تعامل كمتغيرات عشوائية مجهولة التوزيع الإحصائي. وسوف نسعى خلال دراستنا لهذا الباب إلى تحليل العينات العشوائية لتلك المتغيرات من أجل تقدير خصائصها الإحصائية ومؤشراتها (Parameters) . وسوف نستخدم في ذلك التباين (Variance) كمقياس لدقة تلك التقديرات.

٧ -٢ تعريفات وتصنيفات :

عندما نتحدث عن التحليل الإحصائي لمخرجات المحاكاة فلا بد من تحديد ما إذا كان نموذج المحاكاة يمثل نظاماً قابلاً للتوقف أو غير قابل للتوقف. والأنظمة القابلة للتوقف هي كل نظام له وقت محدد لبداية عمله ووقت آخر لنهايته، أو في المقابل له شرط لتوقفه قابل للوقوع (مثل حصول حدث معين عنده يتوقف عمل نموذج المحاكاة). أما الأنظمة غير القابلة للتوقف فهي كل نظام ليس له وقت محدد لبداية أو نهاية عمله ، أو ليس له شرط معين لتوقف عمله . على سبيل المثال ، نظام عمل بنك من الثامنة صباحاً وحتى الخامسة عصراً هو نظام قابل للتوقف ، أما نظام الإسعاف في المستشفى أو نظام الحركة داخل المطار فهو نظام غير قابل للتوقف. ويعتمد اعتبارنا للنظام الخاضع للدراسة على أنه قابل للتوقف أو غير قابل للتوقف

على أهداف دراسة المحاكاة وطبيعة النظام وخصائصه. وعند دراسة المحلل لنظام غير قابل للتوقف فلابد من تحديد فترة معينة يتم خلالها محاكاة النظام ، وكذلك فلابد من تحديد الحالة الأولية للنظام والتي ستبدأ عندها محاكاته. ويكون عادة الغرض من محاكاة مثل هذه الأنظمة هو دراسة سلوكات النظام بعيدة المدى عندما يصبح في حالة مستقرة (Steady State). ويؤثر اختيار الحالة الأولية وفترة أو شرط توقف المحاكاة لمثل هذه الأنظمة تأثيراً كبيراً على سلوكات النظام المشاهدة من خلال المحاكاة .

ومن التصنيفات المهمة لنا قبل شروعنا في دراسة تحليل مخرجات المحاكاة تصنيف نوعية البيانات الخارجة من المحاكاة والمتوفرة لنا للتحليل ، وتصنف هذه البيانات إلى بيانات متقطعة زمنياً وبيانات متصلة زمنياً .

وتُمثل البيانات المتقطعة زمنياً بالقيم المتتابعة :

$$\{y_1, y_2, y_3, \ldots, y_n\}$$

حيث إن القيم $y_1, y_2, y_3, \ldots, y_n$ تمثل عينات لسلسلة متتابعة من المتغيرات العشوائية تمثل أحد مؤشرات الأداء المستخرجة من برنامج المحاكاة ، مثل زمن تأخير كل عميل في نظام الانتظار في الصفوف. كما تُمثل البيانات المتصلة زمنياً بالقيم المتصلة المعطاة بالدالة y والمعرفة كالتالي:

$$\{y(t): 0 \le t \le t'\}$$

حيث تمثل y قيم عينات لعملية عشوائية متصلة في متغير الزمن t تمثل أحد مؤشرات الأداء المستخرجة من برنامج المحاكاة ، مثل طول الصف كدالة في الزمن في نظام انتظار الصفوف (Queuing Systems).

ويلاحظ أنه من الضروري أن تكون القيم المتتالية المستخرجة من برنامج المحاكاة مستقلة إحصائياً حتى يمكننا تطبيق أساليب التحليل الإحصائي التقليدية على تلك النتائج المستخرجة من برنامج المحاكاة. وسوف نرى لاحقاً أنه لتحقيق ذلك سوف نضطر لأخذ هذه القراءات المتتالية لكل من مؤشرات الأداء المطلوبة على أنها مجموعة من متوسطات هذا المؤشر في كل تشغيلة لبرنامج المحاكاة، فتكون بذلك مأخوذة من تشغيلات (Runs) مختلفة لبرنامج المحاكاة بشرط أن يتم استخدام سلاسل مختلفة ومستقلة من الأرقام العشوائية كمدخلات لكل تشغيلة من تشغيلات برنامج المحاكاة ، وأن يتم اختيار الحالة الأولية للنظام التي تبدأ بها كل تشغيلة بصورة مستقلة عن التشغيلة الأخرى.

٣-٧ إيجاد تقديرات لإحصائيات النظام : حالة الأنظمة القابلة للتوقف :

إذا عرفنا (a) كأحد الإحصائيات التي نريد معرفة قيمتها لمجموعة من القراءات العشوائية المتقطعة زمنياً $\{y_1, y_2, y_3, \ldots, y_n\}$ لأحد مؤشرات الأداء (y) والمستخرجة من برنامج المحاكاة، فإنه لا بد من تعريف مُقدر (Estimator) لقيمة هذا الإحصائي (Statistic) هو عبارة عن صيغة حسابية بدلالة القراءات العشوائية المستخرجة تعطي قيمة يعتقد أنها سوف تقترب

أكثر فأكثر من القيمة الحقيقية للإحصائي (a) كلما ارتفع عدد القراءات المستخرجة لمقياس الأداء .

على سبيل المثال ، لو كانت (a) تمثل المتوسط (Mean) ، فإنه يمكن استخدام المقدر ($\hat{a}$) لتقدير قيمة (a) كالتالي :

$$\hat{a} = \frac{1}{n}\sum_{k=1}^{n} y_k$$

ويسمى المُقدر ($\hat{a}$) في هذه الحالة متوسط العينة (Sample mean).

ويعرف المقدر ($\hat{a}$) على أنه غير منحاز إذا كان متوسط المتغير العشوائي ($\hat{a}$) ذا الرمز ($E(\hat{a})$ يساوي (a):

$$E(\hat{a}) = a$$

وفي حال ما إذا كانت القراءات العشوائية متصلة زمنياً ، فإن المقدر ($\hat{a}$) لقيمة الإحصائي (a) الذي يمثل المتوسط لهذه القراءات يعرف كالتالي :

$$\hat{a} = \frac{1}{T}\int_{0}^{T} y(t)dt$$

حيث يسمى($\hat{a}$) في هذه الحالة المتوسط الزمني للقراءات المتصلة والممثلة بالدالة الزمنية y(t) على الفترة [0, T].

لاحظ فيما سبق أن ($\hat{a}$) هو متغير عشوائي نظراً لأنه دالة مجموع لمتغيرات عشوائية (في حالة القراءات المتقطعة زمنياً) ، ولأنه ناتج تكامل

عملية عشوائية (في حالة القراءات المتصلة زمنياً). ونظراً لأن ($\hat{a}$) متغير عشوائي ، فإنه من الطبيعي أن نتساءل عن قيمة التباين (Var) لهذا المتغير لمعرفة إحدى خصائصه الإحصائية المهمة. ولتقدير قيمة التباين للمقدر ($\hat{a}$) والذي سنرمز له بالرمز (Var($\hat{a}$ ، فإننا نحتاج إلى مقدر جديد لقيمة ($\hat{a}$) Var هو $V\hat{a}r(\hat{a})$، والذي قد يكون في حد ذاته منحازاً أو غير منحاز. وفي حال إيجاد مقدر غير منحاز $V\hat{a}r(\hat{a})$، فإنه يمكن الإثبات رياضياً أن المتغير العشوائي (t) والمعرف على أنه :

$$t = \frac{\hat{a} - a}{V\hat{a}r(\hat{a})}$$

سيكون توزيعه الإحصائي بحسب توزيع (t-distribution) ذي المعامل (f) درجة. ومعنى هذه النتيجة أنه من الممكن تعريف فترة تقدير نزعم أن المتوسط (a) سوف يقع فيها – حسب تقديرنا – وبنسبة ثقة قدرها ١٠٠(١ - a)% إذا ما استخدمنا المقدر ($\hat{a}$) ، واستخدمنا لتقدير تباين هذا المقدر $V\hat{a}r(\hat{a})$.

ويمكن صياغة هذه الفترة بصورة رياضية كالتالي :

$$\hat{a} - tf,d/2\sqrt{V\hat{a}r(\hat{a})} \le a \le \hat{a} + tf,d/2\sqrt{V\hat{a}r(\hat{a})}$$

حيث إن $t_{f,d/2}$ هي قيمة المتغير العشوائي (t) من نوع (t-distribution) التي تحقق المعادلة التالية :

$$\Pr\{t \geq t_{f,d/2\sqrt{V\hat{a}r(\hat{a})}}\} = d/2$$

ومـن الممكـن قـراءة قـيمة $t_{f,d/2}$ مـن جـداول الـتوزيع الإحـصائي (-t distribution) والمتوفرة في أغلب كتب ومراجع علم الإحصاء والاحتمالات.

لاحظ أن فترة الثقة المذكورة أعلاه لا تصح إلا في حال استخدام مقدر غير منحاز لقيمة $Var(\hat{a})$ ، وبشرط أن يكون مقدر المتوسط ($\hat{a}$) أيضاً غير منحاز. ولاستيفاء هذين الشرطين ، على المحلل أن يأخذ القراءات y لمؤشر الأداء المستخرج من المحاكاة على شكل متوسطات لهذا المؤشر في كل تشغيلة مستقلة من تشغيلات برنامج المحاكاة (فتكون بذلك كل y تمثل تشغيلة مختلفة ومستقلة لبرنامج المحاكاة،) مع مراعاة أن يتم استخدام سلاسل مختلفة من الأرقام العشوائية كمدخلات لكل تشغيلة ، وأن يتم أيضاً اختيار الحالة الأولية للنظام عند بداية كل تشغيلة بصورة مستقلة .

وفـي حـال اسـتيفاء هـذه الـشروط ، فإنـه بالاسـتناد إلـى نظـرية الـنهاية المركزية المعروفة (Law and Kelton, 2000) يمكن اعتبار أن (S^2/n) هو مقدر غير منحاز لتباين ($\hat{a}$) ، حيث أن (S^2) هو تباين العينة والمعرف كالتالي :

$$S^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{a})^2}{n-1}$$

وفي هذه الحالة ، يمكن كتابة فترة الثقة بدلالة (S^2) كالتالي :

$$\hat{a} - t_{f,d/2}\frac{S}{\sqrt{n}} \leq a \leq \hat{a} + t_{f,d/2}\frac{S}{\sqrt{n}}$$

حيث إن ($\hat{a}$) هو متوسط العينة (Sample Mean) والذي هو مقدر غير منحاز للمتوسط (a). ويسمى المقدار $\frac{S}{\sqrt{n}}$ الخطأ القياسي (Standard Error) .

ولإيجاد فترة الثقة عملياً خلال تحليل مخرجات المحاكاة $y_1, y_2, y_3, \ldots, y_R$ والمستخرجة كمتوسطات مستقلة إحصائياً ومتماثلة التوزيع الإحصائي من (R) تشغيلة لبرنامج المحاكاة ، على المحلل اتباع الإجراء التالي (في حال استخدام عدد معين من التشغيلات R) :

١- احسب قيمة المقدر ($\hat{a}$) (متوسط العينة) للقراءات $y_1, y_2, y_3, \ldots, y_R$ كالتالي :

$$\hat{a} = \frac{1}{R}\sum_{i=1}^{R} y_i$$

٢- احسب قيمة المقدر $V\hat{a}r(\hat{a})$ للقراءات $y_1, y_2, y_3, \ldots, y_R$، كالتالي :

$$V\hat{a}r(\hat{a}) = \frac{S^2}{R} = \frac{1}{R(R-1)}\sum_{i=1}^{R}(y_i - \hat{a})^2$$

٣- احسب فترة الثقة بحسب التالي :

$$\hat{a} - t_{f,d/2\sqrt{V\hat{a}r(\hat{a})}} \le a \le \hat{a} + t_{f,d/2\sqrt{V\hat{a}r(\hat{a})}}$$

حيث $f = R-1$.

مثال :

وفـي حـال أن المطلـوب مـن المحلـل هـو إيجـاد عـدد تـشغيلات بـرنامج المحاكـاة (R) اللازمـة لحصر الخطأ في تقدير (a) في فترة ثقة ذات طول على يمين ويسار قيمة (a) قدره (L) باحتمالية (1-P) ، فعلى المحلل عندئذ اتباع الإجراء التالي :

١- افترض قيمة مبدئية لعدد مرات تشغيل برنامج المحاكاة (R^*) ثم قم بتشغيل البرنامج هذا العدد من المرات بصورة مستقلة.

٢- احسب قيمة تباين العينة في هذه الحالة * (S^2) باستخدام القراءات المستخرجة من المحاكاة بعد تشغيلها (R^*) مرة.

٣- احسب عدد مرات التشغيل (R) للحصول على درجة الثقة المطلوبة باستخدام المتباينة التالية:

$$R \ge \left(\frac{S^{*} t_{d/2, R-1}}{L}\right)^2$$

أي ، خذ (R) كأصغر عدد صحيح أكبر من أو يساوي (R^*) ويحقق المتباينة أعلاه.

٤- قم بتشغيل برنامج المحاكاة ($R-R^*$) مرة إضافية واحصل على القراءات المتبقية.

٥- قم بتكوين فترة الثقة المطلوبة كالتالي:

$$\hat{a} - t_{R-1, d/2}\sqrt{V\hat{a}r(\hat{a})} \le a \le \hat{a} + t_{R-1, d/2}\sqrt{V\hat{a}r(\hat{a})}$$

حيث :

$$\sqrt{V\hat{a}r(\hat{a})} = \frac{S}{\sqrt{R}}$$

مثال :

افترض نظاماً لبنك يحتوي على ثلاث نوافذ للصرافين. يمتد العمل في البنك من الساعة الثامنة صباحاً وحتى الثانية عشرة ظهراً خلال الوردية الأولى . إذا افترضنا أن عملاء البنك يصلون تباعاً وفقاً لتوزيع إحصائي من نوع (Exponential) للمتغير العشوائي الذي يمثل الفترة الزمنية الفاصلة

بين كل عميل والذي يليه . وإذا افترضنا أيضاً أن زمن الخدمة لدى كل صراف يمكن تمثيله بمتغير عشوائي أيضاً من نوع (Exponential) .

احسب فترة الثقة للمقدر (α) لمتوسط زمن التأخر المتوقع للعميل خلال فترة الوردية الأولى وبنسبة ثقة ٩٠%.

الحل:

لحساب متوسط التأخر المتوقع للعميل علينا أن نقوم بإيجاد المتوسط الإحصائي لفترة تأخر كل عميل تمت خدمته خلال فترة الوردية الأولى . فإذا افترضنا أن فترة تأخر العميل رقم (i) في النظام هي (D_i)، فإن (D_i) تمثل متغيراً عشوائياً معيناً . ولعدد (N) عميل تمت خدمتهم خلال فترة الوردية الأولى، فإن المتوسط الإحصائي المتوقع لزمن التأخر يصبح:

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} E(D_i)$$

حيث $E(D_i)$ يمثل المتوسط الإحصائي للمتغير العشوائي (D_i) الذي يعبر عن زمن تأخر العميل رقم (i) في النظام .

فإذا كان N=10 و S^2=0.45 و $\overline{X}_{10} = 2.5$. فإن فترة الثقة للمقدر ($\hat{a}$) لمتوسط زمن التأخر للعميل بنسبة ثقة ٩٠% تصبح :

$$\overline{X}_{10} - t_{9,0.95}\sqrt{\frac{s_{10}^2}{10}} \le \hat{a} \le \overline{X}_{10} + t_{9,0.95}\sqrt{\frac{s_{10}^2}{10}}$$
$$2.5 - 1.82.\sqrt{\frac{0.45}{10}} \le \hat{a} \le 2.5 + 1.82\sqrt{\frac{0.45}{10}}$$
$$2.114 \le \hat{a} \le 2.886$$

٧-٤ إيجاد تقديرات لإحصائيات النظام: حالة الأنظمة غير القابلة للتوقف :

إذا نظرنا إلى نموذج محاكاة لنظام غير قابل للتوقف (Non-terminating) بمدة تشغيل واحدة صدر منها المخرجات $Y_1, Y_2....$ والتي تمثل متغيرات عشوائية تعبر عن القراءات المتتالية لأحد المخرجات، فيعرف المتغير Y على أنه المتغير العشوائي للحالة المستقرة لهذا المخرج إذا كان:

$$F_i(y) \to F(y) = P(Y \le y)$$

كلما كانت $i \to \infty$

وعندما نريد تقدير المتوسط (على سبيل المثال) $E(Y)$ للمخرج الذي يمثله المتغير العشوائي Y ، فإن السبيل إلى ذلك هو من خلال الاعتماد على القراءات $Y_1, Y_2....$، إلا أن ذلك يصطدم بمشكلة وهي أن التوزيع الإحصائي (والخصائص الإحصائية) للمتغيرات $Y_1, Y_2....$ يختلف بشكل عام عن التوزيع الإحصائي للمتغير Y في هذه النوعية من الأنظمة نظراً لعدم استطاعتنا دائماً اختيار الحالة الابتدائية (Initial Conditions) بحيث تمثل الحالة المستقرة (Steady-State) للنظام بحيث تقترب خصائص $Y_1, Y_2....$ من خصائص Y الإحصائية .

وبدون ذلك، فإن المتوسط الحقيقي للمخرج $E(Y)$ لا يمكن تقديره من المتوسط المحسوب باستخدام $Y_1, Y_2 ... Y_m$ ، وليكن $\hat{E}_m(Y)$ ، مهما كانت قيمة i ، ولا يصل إلى القيمة الحقيقية $E(Y)$ إلا عندما تكون $m \to \infty$.

وللتغلب على هذه المشكلة عادة ما يلجأ المحلل إلى أسلوب حذف بعض البيانات الابتدائية من مخرجات النموذج أو ما يعرف أحياناً بمصطلح "إحماء النموذج" (Warming Up the Model). ويقوم هذا الأسلوب على حذف عدد من القراءات التي تخرج في بداية تشغيل نموذج المحاكاة من القراءات الأولى في $Y_1, Y_2 ... Y_m$ والاعتماد على ما بعدها فقط في تقدير قيمة المتوسط الحقيقي $E(Y)$: فإذا كان $\hat{E}_{m,l}(Y)$ هو المتوسط المقدر بهذا الأسلوب ، فيمكن التعبير عنه رياضياً كالتالي (Law & Kelton 2000) :

$$\hat{E}_{m,l}(Y) = \frac{\sum_{i=l+1}^{m} Y_i}{m-l}$$

حيث m هو عدد القراءات الكلي، و l هو طول فترة الإحماء (Warm up Period) . ولأن القراءات المعتمدة في تقدير المتوسط تبعد بمسافة l عن القراءات التي أخذت في بداية المحاكاة، وبالتالي تأثرت أكثر بالحالة الابتدائية ، فإنه يتوقع أن يكون $\hat{E}_{m,l}(Y)$ مقدراً أفضل من $\hat{E}_m(Y)$.

وعادة ما يستخدم أسلوب يسمى أسلوب (Welch) لحساب قيمة l التي تحقق أفضل قيمة محسوبة لـ $\hat{E}(Y_i)$ من حيث قربها من قيمة $E(Y)$ الحقيقية لكل $i > l$. ويعتمد هذا الأسلوب على إجراء عدد $(n > 5)$ من التشغيلات

المستقلة عن بعضها البعض لنموذج المحاكاة بطول m لكل منها ثم تطبيق الخطوات التالية :

١- قم بحساب المتوسط لكل قراءة على جميع التشغيلات، فإذا كانت Y_{ji} هي القراءة رقم i في التشغيلة رقم j ، قم بحساب المتوسط:

$$\bar{Y}_i = \sum_{j=1}^{n} \frac{Y_{ji}}{n} \qquad i = 1,2,...m$$

ويمثل كل $\bar{Y}_i$ متغيراً عشوائياً له المتوسط $E(\bar{Y}_i)$ والتباين $Var(\bar{Y}_i)$. ونظراً لاستقلالية تشغيلات المحاكاة عن بعضها البعض (والذي افترضناها في البداية)، فإن من السهل إثبات أن :

$$E(\bar{Y}_i) = E(Y_i)$$

وأن

$$Var(\bar{Y}_i) = Var(Y_i)$$

٢- بعد ذلك قم بحساب المتوسط المتحرك (Moving Average) $\bar{Y}_i(w)$ حيث w هو طول نافذة المتوسط ، وفقاً للمعادلة التالية (Law & Kelton, 2000) :

$$\bar{Y}_i(w)=\begin{cases}\dfrac{\sum_{S=-w}^{w}\bar{Y}_{i+s}}{2w+1} & i=w+1,\ldots,m-w\\[2ex] \dfrac{\sum_{S=-(i-1)}^{i-1}\bar{Y}_{i+s}}{2i-1} & i=1,\ldots,w\end{cases}$$

٣- قم برسم الشكل البياني للدالة $\bar{Y}_i(w)$ في مقابل $i=1,2,\ldots,m-w$ ثم قم باختيار القيمة l على أنها أقل قيمة i والتي بعدها يبدأ المنحنى البياني للدالة $\bar{Y}_i(w)$ في اتخاذ شكل سطحي أفقي نظراً لتقارب قيم المحور الصادي.

وقد تكرر الخطوة (٣) أعلاه لعدة قيم من w ويختار منها الأقل من مجموعة قيم w التي يظهر عندها المنحنى بشكل سطحي مع تزايد قيمة i.

وبعد اختيار قيمة مناسبة (l) لفترة الإحماء ، يبقى علينا مُهمًّا تقديرُ المتوسط الحقيقي من واقع المتوسط التقديري المحسوب $\hat{E}_{m,l}(Y)$. وسنستعرض هنا طريقة واحدة لعمل ذلك مع العلم بأن هناك طرقًا أخرى عدة بديلة. وتعتمد هذه الطريقة على حساب المتوسط على مستوى جميع قراءات المخرج من بعد فترة الإحماء وفقاً للمعادلة :

$$X_j=\frac{\sum_{i=l+1}^{m}Y_{ji}}{m-1}\qquad j=1,2,\ldots,n$$

وذلك لكل تشغيلة محاكاة j. ونظراً لفرضية استقلالية تشغيلات المحاكاة عن بعضها البعض، فإنه من الممكن إثبات أن المتغيرات

العشوائية X_j ستكون متغيرات مستقلة ومتماثلة التوزيع (IID) ولها المتوسط نفسه تقريباً مثل المتوسط الحقيقي، أي :

$$E(X_j) \approx E(Y)$$

ومن ثم، فإنه بالإمكان حساب فترة ثقة بنسبة ثقة $100(1-a)\%$ للمتوسط الحقيقي $E(Y)$ كالتالي :

$$\overline{X}(n) - t_{n-1,1-\frac{a}{2}} \sqrt{\frac{S^2(n)}{n}} \le E(Y) \le \overline{X}(n) + t_{n-1,1-\frac{a}{2}} \sqrt{\frac{S^2(n)}{n}}$$

حيث n هو عدد التشغيلات المستقلة لنموذج المحاكاة، و $\overline{X}(n)$ هو :

$$\overline{X}(n) = \frac{\sum_{j=1}^{n} X_i}{n}$$

و

$$S^2(n) = \frac{\sum_{j=1}^{n} \left[X_j - \overline{X}(n)\right]^2}{n-1}$$

وحيث $t_{n-1,1-\frac{a}{2}}$ هي قيمة المتغير العشوائي (t) من نوع (t-distribution) ذي

المعامل $(n-1)$ درجة .

لاحظ هنا أن على المحلل تشغيل نموذج المحاكاة لعدد من المرات المستقلة لتقدير قيمة l ، ثم عدد آخر من المرات المستقلة لحساب قيمة المتوسط وفقاً للطريقة المبينة أعلاه.

مثال:

احسب المتوسط المتحرك $\bar{Y}_i(w)$ على افتراض الآتي:

m=6

w=2

$$\bar{Y}_i = i^2$$

الحل:

i=1:

$$= 1\bar{Y}_1 = \bar{Y}_1(2) = \frac{\sum_{S=-0}^{0} \bar{Y}_{1+s}}{2(1)-1}$$

i=2:

$$\overline{Y}_2(2) = \frac{\sum_{S=-1}^{1} \overline{Y}_{2+s}}{2(2)-1} = \frac{1}{3}\{\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3\} = \frac{1}{3}\{1+4+9\} = 4.67$$

i=3:

$$\overline{Y}_3(2) = \frac{\sum_{S=-2}^{2} \overline{Y}_{3+s}}{2(2)+1} = \frac{1}{5}\{\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3 + \overline{Y}_4 + \overline{Y}_5\} =$$

$$\frac{1}{5}\{1+4+9+16+25\} = 11$$

i=4:

$$\overline{Y}_4(2) = \frac{\sum_{S=-2}^{2} \overline{Y}_{4+s}}{2(2)+1} = \frac{1}{5}\{\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3 + \overline{Y}_4 + \overline{Y}_5 + \overline{Y}_6\} =$$

$$\frac{1}{5}\{1+4+9+16+25+36\} = 18$$

الباب الثامن

موضوعات متقدمة في المحاكاة الحاسوبية : المحاكاة المتوازية والموزعة

٨-١ المقدمة :

تعتبر تقنية المحاكاة المتوازية والموزعة من التقنيات الحديثة التي تمكن برنامج المحاكاة من العمل على نظم حاسوبية متوازية أو موزعة مثل الحاسبات ذات المعالجات المتوازية والمتعددة أو الحاسبات الشخصية المربوطة بينياً بشبكة حاسوبية.

ويسعى مصممو برامج المحاكاة إلى تطويعها للعمل على مثل هذه الأنظمة من أجل جني عدد من الفوائد مثل اختصار زمن التنفيذ؛ نظراً لزيادة عدد المعالجات، ودعم قابلية هذه البرامج لتحمل الأعطال ؛ نظراً لوجود معالجات بديلة؛ بالإضافة إلى إمكانية تشغيل برنامج المحاكاة نفسه بشكل متزامن على عدد من الحاسبات الموزعة جغرافياً بهدف إشراك عدد من المستخدمين في هذه المواقع المختلفة في العمل أو التدريب على البرنامج نفسه. وتغطي تطبيقات هذه النوعية من برامج المحاكاة مجالات عدة تشمل المجالات العسكرية والهندسية والتدريبية والترفيهية.

وسنناقش في هذا الباب المبادئ الأساسية للمحاكاة المتوازية بأسلوبيها المتحفظ والمتفائل والتي يمكن من خلالهما بناء نماذج محاكاة تعمل بشكل صحيح وأيضاً بكفاية عالية في البيئات الحاسوبية المتوازية. وسنقوم خلال ذلك بشرح أهم وأشهر الخوارزميات المعروفة في هذا الخصوص مثل خوارزم (Chandy/Misra/Bryant) أو (CMB) المتحفظ ومثل خوارزم (Time Warp) المتفائل.

٨-٢ المحاكاة المتوازية والموزعة : أهدافها وتطبيقاتها :

ظهر الاهتمام بالمحاكاة المتوازية والموزعة بين الباحثين في مجال المحاكاة الحاسوبية منذ بداية التسعينيات الميلادية تقريباً، حيث تنامت الحاجة إلى تمكين برامج المحاكاة من العمل على أنظمة الحاسبات المتوازية والموزعة والتي تتكون من عدد من الحاسبات المربوطة شبكياً ببعضها البعض، والتي يمكن أن تتراوح بين أجهزة مربوطة على شبكة محلية إلى مجموعة من المعالجات المتوازية داخل جهاز واحد. ويهدف تشغيل برامج المحاكاة بهذه الصورة إلى تحقيق عدد من المزايا منها (Fujimoto 2000):

١- اختصار زمن تنفيذ برامج المحاكاة :

وذلك من خلال تقسيم مهمة تنفيذ برنامج المحاكاة إلى عدد من المهام الجزئية التي يمكن تنفيذها على التوازي، والشروع في تنفيذها على معالجات أو أجهزة مختلفة بشكل متواز.

٢- التدريب الافتراضي المشترك عن بعد :

ويقصد بذلك تمكين عدد من المستخدمين المتواجدين في مواقع جغرافية مختلفة من الاشتراك في تدريبات افتراضية تفاعلية باستخدام برنامج محاكاة من مواقعهم المتباعدة بالأسلوب والكفاية نفسها، كما لو كانوا في موقع جغرافي واحد.

٣- احتمال الأخطاء والأعطال:

وذلك من خلال نقل المهام الحسابية للمعالج الذي يتعرض للأعطال إلى معالج آخر لإكمال عمله حتى لا يتوقف تشغيل برنامج المحاكاة.

وفيما يخص تطبيقات المحاكاة المتوازية والموزعة، فهي تشمل الكثير من المجالات التي نذكر منها على سبيل المثال ما يلي:

١- المجالات العسكرية:

وفيها تستخدم هذه النوعية من المحاكاة في تكوين بيئات افتراضية للتدريب المشترك والتدريب عن بعد على العمليات القتالية أو الدفاعية أو الأمنية والتأكد من استيعاب المتدربين للخطط والتكتيكات المعدة مسبقاً. كما أنها تستخدم أيضاً في تقييم هذه الخطط ومدى صلاحيتها وفاعليتها.

٢- المجالات الهندسية :

حيث تستخدم أساليب المحاكاة المتوازية في تسريع أداء برامج المحاكاة الكبيرة التي تستخدم لنمذجة شبكات الاتصالات بمختلف أنواعها والتي عادة ما تتطلب محاكاتها بناء نماذج محاكاة تفصيلية تستغرق زمناً طويلاً في تشغيلها حتى يمكن الحصول على نتائج منها تعين مصمم الشبكات على اختبار الاختيارات التصميمية المختلفة المتاحة له. كما تؤدي المحاكاة المتوازية دوراً مماثلاً في تحسين أداء برامج محاكاة الدوائر المنطقية الإلكترونية كبيرة الحجم والتي تدخل في تصميم وصناعة أجهزة الحاسبات والأجهزة الإلكترونية. فمن المعروف أن محاكاة هذه المكونات الإلكترونية تستخدم باعتبارها وسيلة فعالة للتأكد من صحة وجودة التصميم الهندسي للمكون وللجهاز ككل؛ إلا أنها تستغرق أزمنة طويلة جداً لو تم تشغيلها بشكل تقليدي على معالج واحد. لذا، يستخدم مصممو أنظمة الحاسبات الإلكترونية المحاكاة المتوازية لخفض هذه الأزمنة إلى الحد الذي يسمح بالاستفادة من المحاكاة في تقييم تصميماتهم بدقة.

٨-٣ مبادئ المحاكاة المتوازية :

تتكون المحاكاة المتوازية من عدد من "المحاكيات" التقليدية التي يقوم كل منها بمحاكاة جزء مختلف من أجزاء النظام الحقيقي. ويتطلب ذلك تجزئة النظام الحقيقي إلى أجزاء "مكانية" أي أجزاء تعبر عن مكونات جزئية من الحيز المكاني الإجمالي الذي يشغله النظام الحقيقي. فعلى سبيل المثال : لو أن النظام الحقيقي هو شبكة حاسوبية كبيرة، فقد يتم اختيار الأجزاء على أنها الشبكات المحلية الجزئية (Subnetworks) التي تكون في مجموعها الشبكة الأصلية. ويسمى كل جزء (Space Partition) مكاني من أجزاء النموذج بالعملية الفيزيائية أو الحقيقية (Physical Process). ويتم محاكاة كل عملية فيزيائية بمحاكٍ أو ما يطلق عليه "عملية منطقية" (Logical Process) تمثل نموذجاً جزئياً يحاكي فقط هذا الجزء من أجزاء النظام الحقيقي. وتحتوي العملية المنطقية على جزء من حالة العملية الحقيقية التي تقوم بمحاكاتها بالإضافة إلى عداد يسجل زمن المحاكاة المحلي.

وبناء على هذه الفكرة، فيمكن تشغيل كل عملية منطقية على التوازي على معالج مختلف؛ نظراً لكونها تمثل جزءاً مكانياً مستقلاً من النظام. وتقوم كل عملية منطقية (LP) بمعالجة الأحداث المجدولة زمنياً بها ومن ثم توليد أحداث جديدة يتوقع حدوثها إما في هذا الجزء من النظام أو في جزء مكاني آخر تمثله عملية منطقية أخرى. وهنا يجب على العملية الأولى التي ولدت الحدث إرساله بداخل رسالة إلى العملية الأخرى التي سيقع بها، والتي تقوم عند استلام الحدث بجدولته زمنياً للوقوع مستقبلاً بداخلها عند الزمن الذي يحمله ويسمى الختم الزمني (Timestamp) . وعند وصول زمن هذه العملية

إلى هذا الزمن، يتم معالجة هذا الحدث وتعديل متغيرات الحالة بناء على ذلك لهذه العملية، وكذلك جدولة أحداث أخرى لهذه العملية المنطقية أو لغيرها بالكيفية نفسها .

ويلاحظ هنا أن على كل عملية منطقية معالجة مجموعة من الأحداث التي قد تكون مولدة داخلياً أو خارجياً من قبل عمليات أخرى، ولكن بشرط أن تكون المعالجة بالترتيب غير التنازلي لزمن الأحداث (الختم الزمني) والذي يضمن منطقية النتائج، حيث إن وقوع الأحداث يتم بهذا الترتيب. ولكن على عكس برامج المحاكاة التقليدية التي مرت في هذا الكتاب والتي تستخدم تراكيب بيانات تضمن المعالجة المرتبة للأحداث (مثل: قائمة الأحداث)، فإنه في حالة المعالجة المتوازية للعمليات المنطقية – أي في حالة وجود العمليات المنطقية على معالجات مختلفة، فإنه لا يمكن استخدام الآلية المستخدمة نفسها في المحاكاة التقليدية لهذا الغرض؛ نظراً لعدم وجود ذاكرة مشتركة بين هذه المعالجات (نفترض في هذا النقاش استخدام معالجات غير مشتركة في الذاكرة). ونتيجة لذلك، ينبغي التفكير في حلول جديدة لمشكلة الترتيب الزمني لمعالجة الأحداث في هذه الحالة منعاً لوقوع المعالجة غير المرتبة زمنياً للأحداث وهو ما يعرف بالأخطاء السببية (Causality errors).

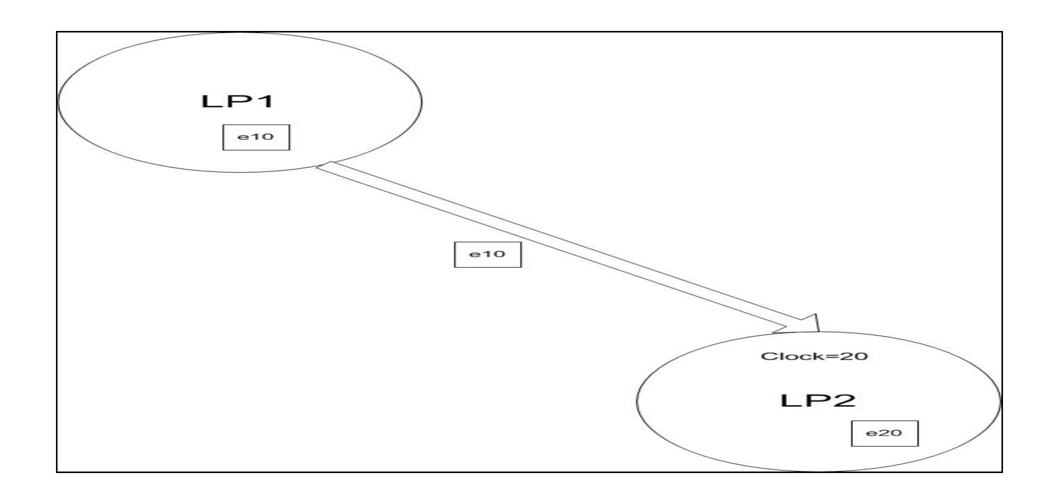
وقد ظهرت عبر سنوات من عمر مجال المحاكاة المتوازية والموزعة الكثير من الحلول والخوارزميات للمشكلة المذكورة، وتسمى مشكلة التزامن (Synchronization Problem). ويمكن تصنيف هذه الحلول إلى صنفين رئيسيين – اعتماداً على الأسلوب المستخدم في حل مشكلة التزامن – هما: صنف الأسلوب المتحفظ (Conservative Approach) وصنف الأسلوب

المتفائل (Optimistic Approach). وسنأتي إلى شرح كل منهما في الفصول القادمة .

ويعتمد الأسلوبان على فكرة المحافظة على مبدأ أن كل عملية منطقية يجب أن تعالج الأحداث بترتيب زمني غير تنازلي، حيث إن أي حياد عن هذا المبدأ قد يتسبب في وقوع الأخطاء السببية لدى هذه العملية المنطقية. كما أنه من المثبت أن الالتزام بهذا المبدأ من قبل جميع العمليات المنطقية في المحاكاة يؤدي إلى الحصول على نتائج صحيحة للمحاكاة المتوازية ككل. ويقصد بصحة النتائج هنا مماثلتها لنتائج المحاكاة غير المتوازية (التقليدية) التي يتم الحصول عليها لو قمنا بتشغيل نفس نموذج المحاكاة على جهاز حاسوبي بمعالج واحد.

وينبغي التنويه هنا بحقيقة : أن الالتزام بمبدأ معالجة كل عملية منطقية لأحداثها بترتيب زمني غير تنازلي هو شرط كاف ولكنه ليس ضروريًّا للحصول على نتائج "صحيحة" من المحاكاة المتوازية. (Fujimoto, 2000). ويفسر ذلك بأن هناك أحداثًا في المحاكاة مستقلة عن بعضها البعض (أي لا توجد علاقة سببية تربط بينها)، لذا فيمكن معالجتها في آن واحد حتى وإن اختلف ختمها الزمني مع عدم الإخلال بصحة نتائج برنامج المحاكاة ككل. ويوضح الشكل رقم (٨-١) مثالاً يمثل حالة وقوع خطأ سببي نتيجة لإرسال العملية المنطقية (LP1) لحدث e_{10} له الختم الزمني (١٠) إلى العملية المنطقية (LP2) والتي عليها أن تعالج الحدث الداخلي لديها e_{20} والمجدول عند الزمن المحلي (٢٠) لديها. فلو أن (LP2) قامت بمعالجة e_{20} قبل استلام e_{10} لنتج عن ذلك معالجة e_{10} بعد e_{20} أي محاكاة حدث من الماضي (زمن ١٠) في الوقت الحاضر (زمن ٢٠

، وهو زمن آخر حدث قامت العملية بمعالجته)، وهو ما يمثل خطأً سببياً. وتتغلب المحاكاة التقليدية على هذه المشكلة باشتراط المعالجة المرتبة بترتيب غير تنازلي لكل من e_5 لدى (LP1) (انظر الشكل رقم ٨-١) و e_{15} و e_{20} . ولكن في حالة المحاكاة المتوازية، فإن هذا الحل غير مقبول؛ لأنه يحول المحاكاة المتوازية إلى تقليدية (تسلسلية أو Sequential) ويفقدها بالتالي إمكانية معالجة الأحداث على التوازي وبالتالي يفقدها مزية السرعة. لذا، تقوم المحاكاة المتوازية بمعالجة الأحداث في كل عملية منطقية وبشكل متواز بشرط استيفائها لقيد السببية المحلية (Local Causality Constraint) والذي ينص على ضرورة معالجة الأحداث محلياً بالترتيب الزمني غير التنازلي .



الشكل رقم (٨-١): مثال على الأخطاء السببية

وفيما يلي نقوم بشرح المفهوم الذي بني عليه الأسلوب المتحفظ في حل مشكلة التزامن.

٨-٤ الأسلوب المتحفظ (Conservative Approach) :

يفترض هذا الأسلوب أن الاتصال بين العمليات المنطقية يتم من خلال قنوات محددة مسبقاً وتتمتع بخاصيتي الاعتمادية والترتيب. كما يفترض أيضاً أن إرسال رسائل الأحداث على أي قناة يتم بترتيب غير تنازلي من حيث الختم الزمني لهذه الأحداث وبالتالي يتم استلامه على الطرف الآخر بهذا الترتيب نفسه. ويتم عند العملية المستلمة تخزين هذه الرسائل في صفوف بأسبقية الوصول (First-in-First-out) لكل قناة اتصال. وتحدد كل عملية منطقية لكل قناة واصلة إليها ساعة (Clock) لهذه القناة تأخذ قيمتها من الختم الزمني لأول حدث في صف الانتظار الخاص بالقناة (أو بالختم الزمني لآخر حدث كان مسبقاً في الصف إذا كان الصف حالياً فارغاً).

ويلاحظ هنا أن الختم الزمني لآخر حدث في صف كل قناة يمثل الحد الأدنى للأختام الزمنية لأي حدث يمكن للعملية المنطقية المستلمة أن تتوقع وصوله مستقبلاً من هذه القناة. ويستنتج من ذلك أنه لو قامت العملية المستلمة بمعالجة الأحداث المخزنة في صف انتظار القناة الواصلة إليها بالترتيب (أي من الزمن الأدنى فصاعداً) طالما وجد في صف هذه القناة رسائل أحداث، فإن ذلك يضمن عدم وقوع أخطاء سببية من معالجة الأحداث القادمة من هذه القناة. وفي حال ما إذا كان هناك أكثر من قناة واصلة إلى العملية المنطقية، تكون معالجة العملية للأحداث في صفوف القنوات القادمة إليها بحيث يتم معالجة الحدث ذي الختم الزمني الأصغر من القناة الأولى. فإذا وجد عقبه أحداثاً أخرى في الصف نفسه أو في صفوف القنوات الأخرى بختم زمني مساو يتم حذف ذلك الحدث من صفه ومعالجته. وبعد معالجة

كل حدث يتم تحديث قيمة العداد الزمني (الساعة) للقناة التي تم استلامه منها إلى قيمة تساوي الختم الزمني لذلك الحدث. بعد ذلك يتم معالجة الحدث ذي الختم الزمني الأصغر بين الأحداث المخزنة على رأس كل صف من صفوف انتظار القنوات الواصلة والذي يلي في القيمة قيمة الساعة لآخر قناة تم الاستلام منها، وهكذا. وتمتنع العملية عن معالجة هذا الحدث التالي في حالة ما إذا كان آخر حدث تمت معالجته من إحدى القنوات القادمة للعملية هو أيضاً آخر حدث مخزن في صف انتظار تلك القناة. فعندئذ تكون هناك احتمالية لوصول حدث آخر لاحقاً على القناة نفسها بقيمة الختم الزمني السابق نفسه أي بقيمة ساعة هذه القناة نفسها.

وبناء على هذه القواعد تضمن العملية تفادي الوقوع في الأخطاء السببية باتخاذ هذا الأسلوب "المتحفظ" في معالجة الأحداث الواصلة إليها . ولكن يعيب هذه القواعد أنها قد تؤدي في كثير من الأحيان إلى حدوث حلقة مغلقة من صفوف انتظار القنوات الفارغة، مما يشل حركة برنامج المحاكاة ويؤدي إلى حالة توقف (Deadlock) لا تستطيع المحاكاة معها التقدم إلى الأمام في عملها على الرغم من وجود أحداث غير معالجة في صفوف القنوات الواصلة إلى العمليات المنطقية.

ولعلاج هذا الوضع يجب على العملية المنطقية الواحدة أن تبلغ كلاً من العمليات الأخرى التي تقوم بالاستقبال من هذه العملية بالحد الأدنى للختم الزمني للرسائل التي سوف تقوم بإرسالها مستقبلاً إلى كل منها. وتتم هذه العملية في الأسلوب المتحفظ باستخدام ما يعرف بالرسائل الباطلة (Null messages) ، وهي عبارة عن رسائل فارغة من المحتوى تقوم فقط بمهمة

تحكمية تتعلق بمنع حدوث حالات توقف المحاكاة المشار إليها سابقاً (Deadlocks) عن طريق إبلاغ العملية الواقعة على الطرف المستلم بالحد الأدنى للختم الزمني للرسائل التي سوف ترسل إليها في المستقبل. ويتم حساب هذه القيمة لكل قناة خارجة من العملية المنطقية الواحدة عن طريق الاستفادة من قيمة ساعة القناة الواصلة إلى العملية، وذلك لكل قناة واصلة. فكما ذكرنا سابقاً، فان قيمة هذه الساعة (Clock) تمثل حداً أدنى للختم الزمني للأحداث التي سيتم استقبالها من هذه القناة (نظراً لأن الأحداث مرتبة بترتيب غير تنازلي).

فبالتالي، لو أخذنا أصغر قيمة من قيم ساعات القنوات الواصلة إلى العملية لكان ذلك حداً أدنى لقيمة الختم الزمني لأي رسالة سيتم استقبالها من قبل هذه العملية مستقبلاً من أي من القنوات الواصلة إليها. فإذا كان لدينا علم كاف بالنموذج الذي تمثله هذه العملية بحيث يمكن تحديد الحد الأدنى لزمن المحاكاة الذي سيضاف إلى قيمة الختم الزمني لأي حدث يتم معالجته بواسطة هذه العملية (أي الذي يمثل مدة العمليات التي تجرى من قبل هذه العملية على الحدث في زمن المحاكاة)، فإنه بالإمكان عندئذ تقدير الحد الأدنى لقيمة الختم الزمني لرسائل الأحداث الخارجة من هذه العملية إلى أي عملية أخرى على أنه: الحد الأدنى لقيمة ساعات القنوات الواصلة للعملية + الحد الأدنى لقيمة زمن معالجة الحدث الحالي. ويعرف هذا الأخير باسم البعد المستقبلي (Lookahead). وبناء على هذه الفكرة، فإنه على كل عملية أن تقوم بإرسال رسالة (Null) على كل قناة خارجة فور انتهائها من معالجة كل حدث. ويعرف الخوارزم المتحفظ الذي تم وصفه هنا باسم خوارزم

(Chandy/Misra/Bryant) أو (CMB). ويلاحظ هنا أن أداء هذا الخوارزم يرتبط بشدة بحجم البعد المستقبلي (Lookahead) بحيث أن الأداء يتدهور بسرعة كلما صغر حجم (Lookahead) للعملية ككل، حتى يصل إلى الفشل التام في حال وجود حلقة كاملة من العمليات ذات البعد المستقبلي (Lookahead) بقيمة صفر. لذا، فتقدير قيمة (Lookahead) للنموذج وعملياته له أثر كبير في إنجاح الأسلوب المتحفظ المبني على الخوارزم الحالي (CMB) أو إفشاله (بسبب تضخم عدد الرسائل الباطلة المطلوبة لتحريك المحاكاة قدماً).

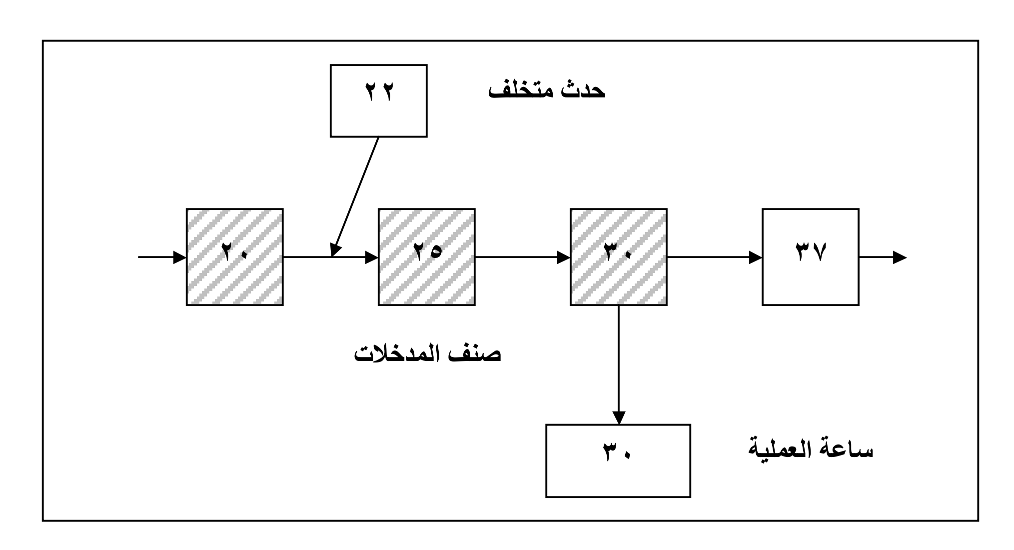
ويجدر بالذكر أن الأسلوب المتحفظ ينطوي على بدائل أخرى كثيرة لخوارزم (CMB) والذي يقوم بتفادي حالات التوقف (Deadlocks). فهناك على سبيل المثال خوارزميات الاكتشاف والاسترجاع (Detection and Recovery) التي تسمح بوقوع (Deadlock) لكنها تقوم عند اكتشافه بإطلاق ما يلزم من الإجراءات للاسترجاع منه (Fujimoto, 2000). وتتميز هذه الخوارزميات بأنها تستطيع التعامل مع النماذج ذات البعد المستقبلي (Lookahead) ذي القيمة صفر، على العكس من (CMB) إلا أنها أكثر تعقيداً. ولكن بصفة عامة، يعيب الخوارزميات المتحفظة تأثر أدائها بقيمة (Lookahead) المتاح في النموذج الذي يؤثر على كيفية تعريف المحلل للنموذج بحيث يحصل على أكبر قدر من (Lookahead) وهو ما يؤدي إلى برامج غير مرنة وصعبة الصيانة والتعديل.

٨-٥ الأسلوب المتفائل (Optimistic Approach) :

يختلف الأسلوب المتفائل عن الأسلوب المتحفظ بشكل جذري، حيث يسمح هذا الأسلوب للمحاكاة المتوازية بمخالفة شرط السببية المحلية (Local Causality). ولكن عند اكتشاف هذه المخالفة تقوم المحاكاة بعملية استرجاع (Recovery) من هذا الخطأ السببي، ثم تستأنف بعد ذلك ما تبقى من عملها. ويعتبر خوارزم استدارة الزمني (Time Warp) الأكثر شهرة واستخداماً من بين خوارزميات الأسلوب المتفائل. ويفترض هذا الخوارزم نفس افتراضات الأسلوب المتحفظ للمحاكاة المتوازية من حيث تجزئة النموذج إلى عمليات منطقية لا تشترك في متغيرات الحالة لكل منها ساعة محاكاة محلية وتتراسل باستخدام رسائل أحداث فيما بينها. إضافة لذلك، فإن خوارزم استدارة الزمن لا يفترض وجود قنوات اتصال ثابتة بين العمليات المنطقية، بل يفترض نمط اتصال عديم القنوات (Connectionless) بين عمليات النموذج.

ويتطلب هذا الخوارزم من كل عملية منطقية الاحتفاظ بقائمة مرتبة زمنياً للأحداث الواردة يسمى صف المدخلات (Input Queue) وقائمة أخرى لتخزين الأحداث التي تم معالجتها من بين أحداث صف المدخلات. وتحتاج العملية المنطقية إلى الاحتفاظ في صف المخرجات (Output Queue) بالرسائل الخارجية الناتجة منها لبعض الوقت للتأكد من عدم اكتشاف حدوث مخالفة لشرط السببية المحلية من قبلها يتطلب إجراء عملية استرجاع. ويحدث هذا الاكتشاف عندما يرد إلى العملية رسالة حدث تحمل ختماً زمنياً أقل من الزمن المحلي المسجل في ساعة هذه العملية (والذي يعبر عن الختم الزمني لآخر حدث من تلك الأحداث المخزنة في صف المدخلات تمت معالجته بداخل هذه العملية). فلو وصل مثل هذا الحدث ذو الزمن الأقل من

الساعة المحلية للعملية – يطلق عليه مسمى "الحدث المتخلف" أو (Straggler event)- فإن العملية تعرف بأن هناك تجاوزاً قد حدث (بسبب إفراطها في التفاؤل) لشرط السببية، وعليها الآن إطلاق عملية التفاف للوراء (Rollback) حتى يمكنها الاسترجاع من هذا الخطأ. ويبين الشكل رقم (٨-٢) حالة وصول حدث متخلف بختم زمني قدرة (٢٢) إلى عملية بلغت قيمة ساعتها (٣٠).



الشكل رقم (٨–٢) : حالة وصول حدث متخلف إلى إحدى العمليات المنطقية (LP

ومصدر الخطأ في مثل هذه الحالات هو المعالجة غير المرتبة زمنياً (بشكل غير تنازلي) لمعالجة الأحداث، حيث يدل ورود الحدث المتخلف بختم زمني قيمته أقل من ساعة العملية على أن معالجة الأحداث لدى هذه العملية (وما يتبعها من تحديث لحالة العملية) قد تجاوزت هذا الحدث المتخلف في الترتيب، وبالتالي على العملية أن تعود إلى الوراء في زمن المحاكاة إلى النقطة المناسبة – أي إلى الحدث المخزن في صف المدخلات

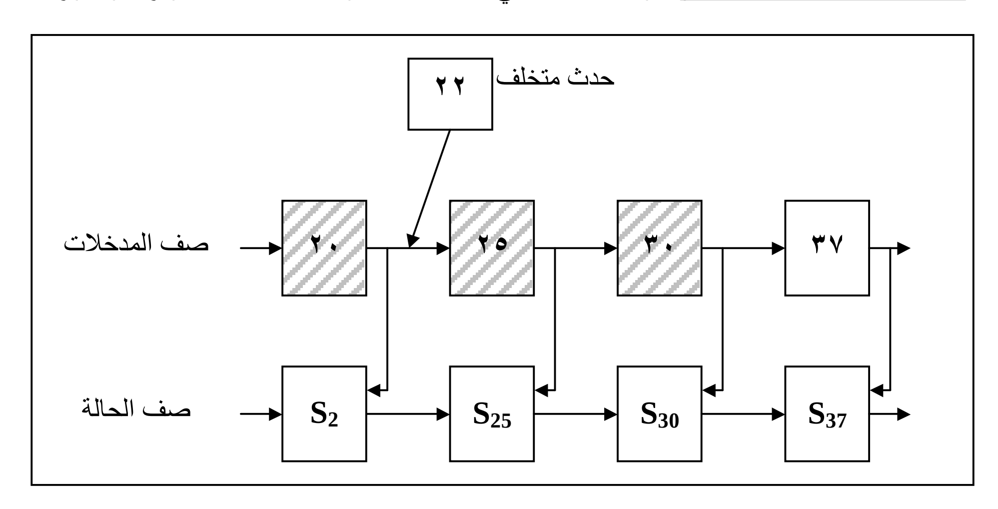
ولـه الختم الزمنـي السابق مباشـرة لقيمة الختم الزمني للحدث المتخلف – ثم البدء فـي معالجـة الحدث المتخلف بعد إسقاطه في مكانه الصحيح في صف المدخلات، ثم مواصلة المعالجـة لمـا يليه زمنياً من الأحداث المخزنة فـي صـف المدخلات. وتسمى عملية الاسترجاع هذه بعملية الالتفاف إلى الوراء (Rollback).

وبالـرجوع إلـى الـشكل رقـم (٨-٢)، فـإن عملـية الالـتفاف إلـى الـوراء ستتطلب إعادة ساعة العملية إلى الوراء إلى الزمن (٢٠) وإسقاط الحدث المتخلف ذي الختم الزمني (٢٢) في مكانه بصف الانتظار، ثم الشروع في معالجـة الحـدث المـتخلف (٢٢) ثـم الحـدث (٢٥) وهكـذا. ويلاحـظ هـنا أن العملية المنطقية كانت قد أنهت معالجة الأحداث ذات الختم الزمني الأقل من قيمة الساعة وأكبر من قيمة زمن الحدث المتخلف [الأحداث ٢٥ و٣٠ في الـشكل رقـم (٨-٢)] ، ممـا يعنـي قيام العملـية بـتعديل قيم متغيـرات الحالـة الخاصة بها نتيجة لمعالجة كل حدث من هذه الأحداث، بالإضافة إلى احتمال إرسال رسائل إلى عمليات منطقية أخرى نتيجة أيضاً لهذه المعالجة. ولكن نظراً لقيام العملية المنطقية بعمل التفاف إلى الوراء بسبب الحدث المتخلف بمعنـى أنهـا قـررت التـراجع عـن معالجـتها الـسابقة لهـذه المجمـوعة مـن الأحـداث، فـإن ذلـك يملـي علـيها الآن مهمـة جديـدة وهـي: إلغـاء أثـر هـذه المعالجة السابقة لهذه الأحداث.

ومـن الـنقاش الـسابق نـرى أن إلغـاء أثـر المعالجـة هـنا يعنـي مـا يلـي: استرجاع حالـة العملية المنطقية التـي كانـت عليها بعد معالجتها الحدث ذي الختم الزمنـي الـذي يسبق مباشـرة الحـدث المـتخلف فـي التـرتيب فـي صـف

المدخلات، وكذلك إلغاء أثر إرسال أي رسالة تم إرسالها إلى عمليات أخرى في المحاكاة المتوازية نتيجة للمعالجة الخاطئة لمجموعة الأحداث التي ألغاها وصول الحدث المتخلف. وكما نتوقع هنا، فإن لهذه الخطوات اللازمة لإلغاء الأثر متطلباتها الخاصة من العملية المنطقية. فبالنسبة لاسترجاع الحالة القديمة، فإن ذلك يتطلب من كل عملية نسخ متجه متغيرات الحالة الخاص بها بصفة دورية إلى الذاكرة، حيث تعرف هذه العملية بالنسخ الدوري للحالة (Periodic State-Saving) . ويكون النسخ هنا إلى صف خاص يلحق بكل عملية يسمى صف الحالة (State Queue).

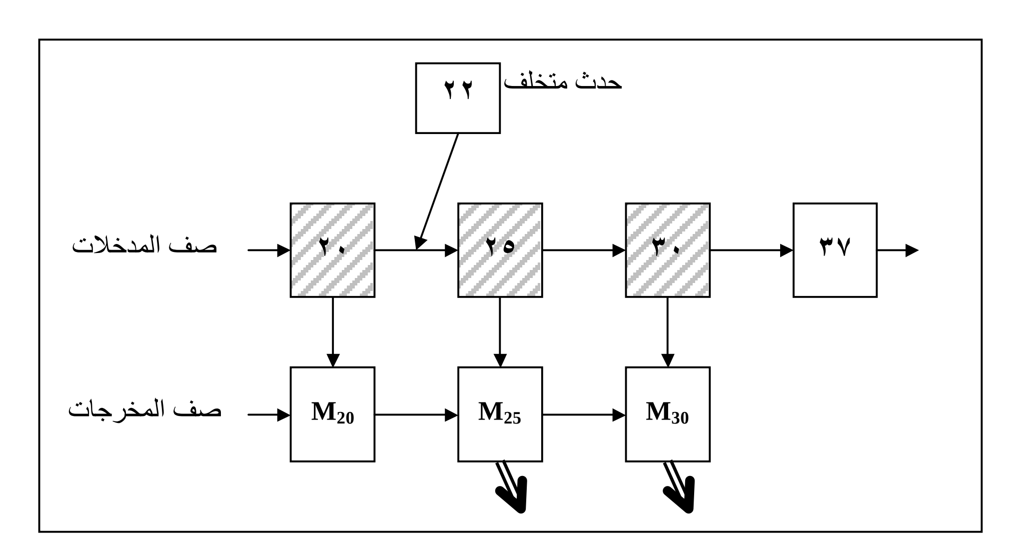
ويضبط معدل الحفظ إلى صف الحالة بحسب خصائص النموذج، فالمعدل العالي للحفظ يعني احتياجات تخزينية أكبر ووقت حفظ أكبر .. بينما معدل الحفظ الأقل يعني قلة عدد الحالات المحفوظة وبالتالي احتياج العملية للرجوع إلى الوراء لمسافة أبعد واسترجاع حالة أقدم زمناً من الزمن المطلوب منها الرجوع إليه بموجب عملية الالتفاف إلى الوراء، ثم إعادة التنفيذ من هذا الزمن الأقدم فصاعداً حتى تستطيع استعادة حالتها عند الزمن المطلوب الالتفاف إلى الوراء إليه، ويمثل الشكل رقم (٨-٣) مثالاً يوضح كيفية حفظ الحالة وفقاً لأسلوب الحفظ الدوري.



الشكل رقم (٨-٣) : تخزين حالة العملية (LP) دورياً بمعدل =١ (أي بعد كل

أما ما يخص إلغاء أثر إرسال الرسائل الخارجية إلى العمليات المنطقية الأخرى، فقد سبق الإشارة إلى أن خوارزم استدارة الزمن يتطلب من كل عملية منطقية (LP) الاحتفاظ بصف للمخرجات يتم فيه تخزين نسخ من الرسائل الخارجية الناتجة عن معالجتها للأحداث الموجودة في صف المدخلات، وذلك لأمر معين تحسباً لحدوث خطأ سببي يتطلب الالتفاف إلى الوراء. فإذا حدث ذلك، تقوم العملية الملتفة إلى الوراء باستخدام الرسائل المخزنة لديها في صف المخرجات لإرسال رسائل مضادة (Antimessages) إلى نفس العمليات الخارجية التي أرسل إليها سابقاً رسائل خاطئة تحتاج إلى الإلغاء. فإذا استلمت إحدى العمليات رسالة معينة ثم استلمت لاحقاً رسالة مضادة لها، فإنها تقوم بإزالة الرسالتين من صف انتظارها، فيحدث بذلك الإلغاء للرسالة الأولى وهو المطلوب. والرسالة المضادة ما هي إلا نسخة طبق الأصل من الرسالة الأصلية؛ إلا أنها تحمل علامة تدل على أنها رسالة إلغاء أو رسالة مضادة. وبهذه الكيفية ما على العملية المنطقية لإلغاء رسالة

خارجية سبق إرسالها سوى استرجاع نسخة هذه الرسالة المخزنة في صف المخرجات وتعليمها بعلامة الرسالة المضادة ثم إرسالها إلى الجهة نفسها التي أرسلت لها الرسالة الأصلية.



الشكل رقم (٨-٤) : كيفية تخزين الرسائل المضادة M_{20} و M_{25} و M_{30} للرسائل المرسلة نتيجة معالجة الأحداث ٢٠ و ٢٥ و ٣٠ وكيفية إرسال الرسائل المضادة M_{20} و M_{30} عند

ويوضح الشكل رقم (٨-٤) الكيفية التي بها تخزن نسخ الرسائل المخرجة في صف المخرجات، والكيفية التي بها ترسل الرسالة المضادة للرسائل المخرجة من جراء معالجة الأحداث ذات الختم الزمني ٢٥ و ٣٠. لاحظ هنا أن عملية إلغاء أثر الرسائل المرسلة للعمليات المنطقية الأخرى قد تتضمن إجراء المزيد من عمليات الالتفاف إلى الوراء لهذه العمليات الأخرى في حال ما إذا قامت بالفعل العملية المنطقية المستلمة للرسالة الخاطئة بمعالجتها. ففي هذه الحالة يجب على هذه العملية أن تجري التفافاً

إلى الوراء هي الأخرى قبل أن تستطيع إلغاء الرسالة الخاطئة التي وصلتها وخزنتها في صف مدخلاتها بموجب استلامها للرسالة المضادة لتلك الرسالة. لاحظ أيضاً أن هذه العملية قد تتكرر مرات عدة مما ينذر بحدوث موجة من الالتفافات المتكررة إلى الوراء على مستوى المحاكاة ككل، وهو ما يعتبر أحد أبرز عيوب خوارزم استدارة الزمن.

ولتقليل فرص حدوث هذه الظاهرة، عادة يتم جدولة تشغيل العمليات المنطقية الواقعة على معالج واحد بحيث يتم تقييم صفوف مدخلات جميع العمليات على المعالج ومنح الدور في التشغيل للعملية التي لديها الحدث المطلوب معالجته الآن وختمه الزمني هو الأصغر من بين أزمنة أمثاله من الأحداث الواقعة على رأس صفوف المدخلات لأي من العمليات المنطقية على هذا المعالج. ويستند ذلك إلى التوقع بأن العملية التي لديها هذا الحدث ذو الزمن الأصغر ستكون الأقل احتمالا لأن تضطر إلى الالتفاف إلى الوراء من قبل رسائل تصلها في المستقبل المحاكي، حيث يمكن إثبات أن الحدث ذا الزمن الأصغر على مستوى المحاكاة ككل لا يمكن أن يحدث له التفاف إلى الوراء (Fujimoto, 2000).

وأخيراً يبقى علينا بعد ما تقدم عن خوارزم استدارة الزمن أن نناقش التساؤل التالي: ما هو الحد الأدنى الزمني الذي لا يمكن للمحاكاة المتوازية تحت خوارزم استدارة الزمن أن تلتفت إلى الوراء زمنياً دونه؟ وتكمن أهمية هذا السؤال في الحاجة عند أي لحظة إلى وجود التزام بعدم العودة زمنياً إلى زمن محاكاة أقل من حد معين (T) حتى يمكن للمحاكاة المتوازية أن تقوم بتأدية بعض المهام التي لا يمكن الاسترجاع منها في حال حدوث التفاف إلى الوراء مثل عمليات المدخلات والمخرجات (I/O) ، وكذلك للقيام بمهمة استعادة مساحات الذاكرة التي

لم تعد المحاكاة محتاجة لها، وهو ما يعرف بمصطلح جمع الأحافير (Fossil Collection) ، فبدون معرفة هذا الزمن (T) لا تستطيع المحاكاة ككل الالتزام في إجراء عمليات الإدخال والإخراج ولا جمع الأحافير (والذي قد يؤدي إلى امتلاء الذاكرة قبل انتهاء تشغيل المحاكاة).

ويعرف هذا الزمن (T) باسم الزمن الافتراضي الشامل (Global Virtual Time) أو اختصاراً (GVT). ويحسب هذا الزمن عند أي لحظة على أنه القيمة الأصغر من بين قيم الختم الزمني عند الاستلام لجميع الأحداث غير المعالجة والمعالجة جزئياً والرسائل المضادة في نظام المحاكاة المتوازية. ويشمل ذلك الرسائل التي في حالة انتقال من معالج لآخر حيث تحمل رسائل غير معالجة. ويمكن بعد حساب هذه القيمة (GVT) أن تقوم المحاكاة المتوازية بإجراء عمليات الإدخال والإخراج (I/O) وباستعادة الذاكرة المستخدمة لأحداث ذات ختم زمني أقل من (GVT) وجميع ما يتعلق بها من رسائل مضادة ومعلومات حالة. وعادة ما يتم حساب قيمة (GVT) على مستوى المحاكاة المتوازية ككل بشكل دوري بواسطة خوارزم خاص أو عندما تدعو الحاجة لذلك (على سبيل المثال: عند نفاد الذاكرة). وهناك الكثير من تلك الخوارزميات المعروفة لحساب قيمة (GVT) والتي تجتهد في حساب هذه القيمة بشكل سريع وفعال، مع العلم بحاجة هذه الخوارزميات إلى الأخذ في الاعتبار الرسائل الموجودة في حالة انتقال بين معالجين لحظة حساب قيمة (GVT) وإلى إيقاف العمل بالمحاكاة المتوازية مؤقتاً خلال فترة حساب قيمة (GVT).

ويجدر في النهاية الذكر بأن ما تم وصفه في هذا الفصل هو خوارزم استدارة الزمن (Time Warp) الأساس. وهذا الخوارزم – بلا شك – يتفوق في أدائه وسرعته في كثير من التطبيقات والنماذج على أمثاله من

الخوارزميات المتحفظة؛ إلا أنه في حالات أخرى يعاني من ضعف قد يصل إلى انهيار في الأداء (على سبيل المثال: في حالة موجات الالتفاف إلى الوراء المتكررة). لذا، تذخر الأدبيات في هذا المجال بعشرات من الأساليب المبتكرة والخوارزميات المعدلة أو البديلة لمعالجة عيوب خوارزم استدارة الزمني؛ إلا أن المقام لا يتسع هنا للحديث عن هذه البدائل البحثية والتي يمكن الاطلاع على بعضها في (Fujimoto,2000).

المصادر والمراجع

أولاً – المراجع العربية :

١- رمضان، حسام محمد و الضلعان ، عبد الله محمد. "تطبيقات المحاكاة الحاسوبية في التخطيط والتدريب على إدارة الكوارث والأزمات"، **مجلة البحوث الأمنية**، العدد (٢٢) .- الرياض، كلية الملك فهد الأمنية ، ١٤٢٣هـ.

٢- رمضان، حسام محمد وآخرون . **التقرير الختامي لمشروع التوسع في استخدام نماذج المحاكاة لدراسة نظام نقل الحجاج بالرحلات الترددية لموسم حج ١٤٢٤هـ** ". وزارة الحج- المجلس التنسيقي للنقل بالرحلات الترددية وجمعية الحاسبات السعودية ١٤٢٥هـ.

ثانيًا – المراجع الإنجليزية :

1- Banks, J., J. S. Carson, II, B. L. Nelson, and D. M. Nicol. Discrete-Event System Simulation. New Jersey: Prentice-Hall, 2001 .

2- Fishwick, P. A. Simulation Model Design and Execution: Building Digital Worlds. New Jersey: Prentice-Hall, 1995 .

3- Fujimoto, R. M. Parallel and Distributed Simulation Systems. New York: John Wiley & Sons, 2000 .

4- Hill, D. R. C. Object-Oriented Analysis and Simulation. Reading, MA: Addison-Wesley, 1996 .

5- Hoover, S. V. and Perry, R. F. Simulation: A Problem-Solving Approach. Reading, , 1990 .

6- Kelton, W. D., Sadowski, R. P. and Sadowski, D. A. Simulation with Arena. Singapore: McGraw-Hill, 2001 .

7- Kuhl, F., R. Weatherly, and J. Dahmann. Creating Computer Simulation Systems: An Introduction to the High Level Architecture. New Jersey: Prentice-Hall, 1999 .

8- Laguna, M. and J. Marklund Business Process Modeling, Simulation, and Design. New Jersey: Pearson Prentice Hall, 2005 .

9- Law, A. M., and W. D. Kelton Simulation Modeling and Analysis. Singapore: McGraw-Hill, 2000 .

10- Mallach, E. G. Decision Support and Data Warehouse Systems. Singapore: McGraw-Hill, 2000 .

11- Papoulis, A. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1984 .

12- Pidd, M. Computer Simulation in Management Science. West Sussex, England: John Wiley & Sons, 1998 .

13- Rockwell Software. *Arena* STANDARD**:** User's Guide. Wisconsin: Rockwell Software, 2003 .

14- Soliman, H. M. "Parallel and Distributed Simulation: Methodologies and Techniques". *Journal of King Saud University-Computer and Information Sciences*, Vol. 10, No. 3, pp. 27-51, Riyadh, 1998 .

15- Walpole, R. E. and R. H. Myers. Probability and Statistics for Engineers and Scientists. New York: Macmillan Company, 1985

ملحق (أ)

مراجعة لأساسيات الإحصاء والاحتمالات

١- المقدمة :

تعتمد دراسات المحاكاة على علم الإحصاء والاحتمالات بشكل مكثف في تحليل عمل النظم العشوائية والتأكد من صحة نماذجها ولاختيار التوزيعات الإحصائية للمدخلات وتوليد الأرقام العشوائية وتحليل المخرجات. ومن هذا المنطلق، يكتسب الإحصاء والاحتمالات أهمية كبيرة في هذه الدراسات، مما يدعونا إلى إفراد هذا الملحق لمراجعة أهم المبادئ والمفاهيم الإحصائية ذات الفائدة والاستخدام المباشر في الموضوعات المعروضة في هذا الكتاب.

٢- المتغيرات العشوائية :

يعرف المتغير العشوائي (X) على أنه دالة تناظر بين كل نقطة في فراغ العينة وبين مجموعة الأعداد الحقيقية بحيث أن لكل نقطة في فراغ العينة w فإن هناك قيمة $X(w) \le a$ تناظرها ، حيث أن a هو أي عدد حقيقي. ويمكن للمتغير العشوائي أن يكون متقطعاً (Discrete) أو متصلاً (Continuous) . ويعرف المتغير العشوائي $X(w)$ على أنه متقطع إذا كانت جميع القيم التي يمكن أن يأخذها $X(w)$ متقطعة. ويمكن في هذه الحالة تعريف "دالة الكثافة الاحتمالية" أو "دالة الكتلة" (Probability Mass Function) أو (Probability Density Function) على أنها :

$$f_x(x_i) \stackrel{\Delta}{=} \Pr\{X(w_i) = x_i(w_i)\}$$
$$= \Pr(X = x_i) = p_i$$

حيث $\Pr()$ هي الاحتمالية .

ويجب لدالة الكثافة الاحتمالية أن تحقق الشروط التالية :

$$1)\ f_x\ (x_i) \geq 0$$

$$2)\ \sum_{i=-\infty}^{\infty} f_x(x_i) = 1$$

وتعرف دالـة الـتوزيع الاحتمالـي (Probability Distribution Function) $F_x(x(w_i))$ للمتغير العشوائي $X(w_i)$ على أنها :

$$\begin{aligned} F_x(x_i) &\stackrel{\Delta}{=} \Pr\{X(w_i) \leq x_i(w_i)\} \\ &= \Pr\{X \leq x_i\} \\ &= \sum_{x \leq x_i} f_x(x_i) = \sum_{i:\{X \leq x_i\}} P_i \end{aligned}$$

مثال:

إذا أعطيت المتغير العشوائي X الذي له القيم :

- 1, 0, 1

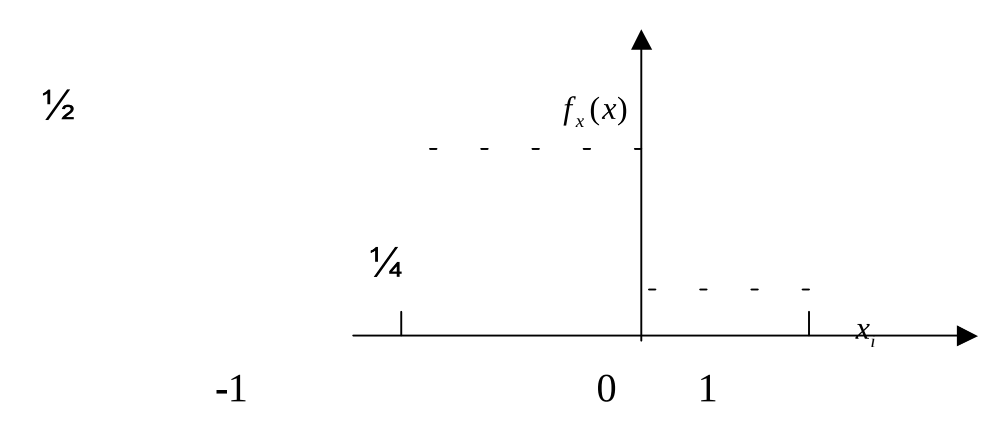
باحتماليات ¼ ، ¼ ، ½ على الترتيب .

فإن شكل دالة الكثافة الاحتمالية سيكون كالتالي :

$$f_x(x) = \begin{cases} \frac{1}{2}, & x = -1 \\ \frac{1}{4}, & x = 0 \\ \frac{1}{4}, & x = 1 \end{cases}$$

0 ، Otherwise

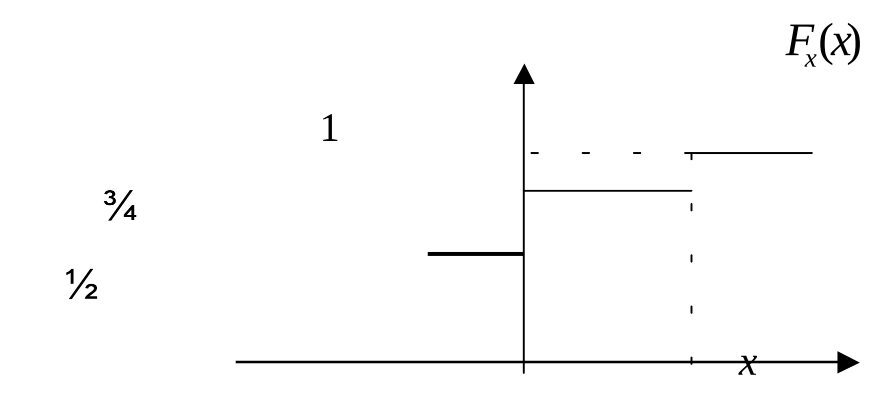
وعند رسم $f_x(x)$ ستبدو هكذا :



بينما يمكن حساب $F_x(x)$ كالتالي :

$$F_x(x) = \begin{cases} 0 \ , & x < -1 \\ \frac{1}{2} \ , & -1 \le x < 0 \\ \frac{3}{4} \ , & 0 \le x < 1 \\ 1 \ , & x \ge 1 \end{cases}$$

وعند رسم $F_x(x)$ ستبدو هكذا :



-1 0 1

ويعرف المتغير العشوائي على أنه متصل إذا كان :

١) $P_r(X = x) = 0$

لكل القيم x

٢) توجد دالة $f_x(x)$ بحيث :

$$\Pr\left\{x \le X \le x + \Delta x) = \int_x^{x+\Delta x} f_x(a)da\right\}$$

ويطلق على الدالة $f_x(x)$ دالة الكثافة الاحتمالية (Probability Density Function) وتحقق الخصائص التالية :

١) $f_r(x) \le 0$

٢) $\int_{-\infty}^{\infty} f_x(x)dx = 1$

ويمكن تعريف دالة التوزيع الاحتمالي في هذه الحالة على أنها الدالة $F_x(x)$ بحيث :

$$F_x(x) \stackrel{\Delta}{=} \Pr\{X \le x\} = \int_{-\infty}^{x} f_x(a)da$$

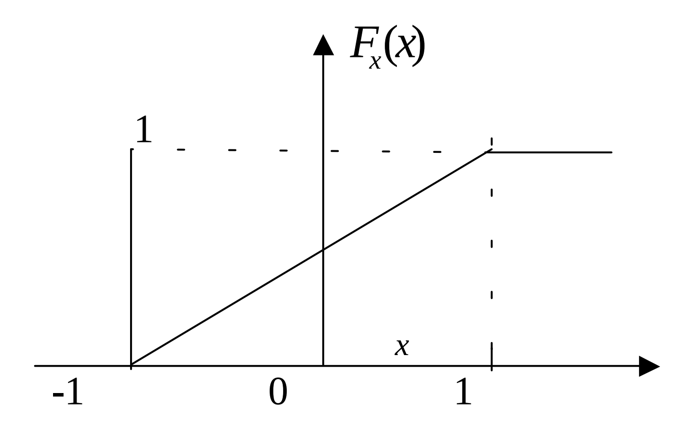
ويمكن هنا إثبات أن

$$f_x(x) = \frac{d}{dx}F_x(x)$$

مثال: إذا أخذنا المتغير العشوائي x الذي له دالة التوزيع $F_x(x)$ المبينة أدناه :

$$F_x(x) = \begin{cases} 0\ , & x < -1 \\ \frac{1}{2}\ (x+1)\ , & -1 \le x < 1 \\ 1\ , & x \ge 1 \end{cases}$$

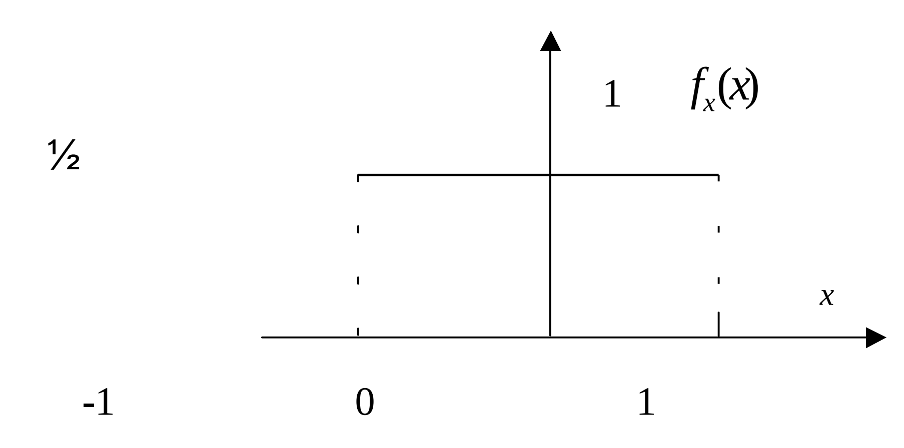
والتي عند رسمها تبدو هكذا :



فإذا أردنا إيجاد دالة الكثافة الاحتمالية ، فيمكننا حساب ما يلي :

$$f_x(x) = \frac{d}{dx} F_x(x) = \begin{cases} \frac{1}{2} & -1 \le x < 1 \\ 0 & \text{Otherwise} \end{cases}$$

ويمكن رسم $f_x(x)$ لتبدو هكذا :



ومن خصائص دالة التوزيع الاحتمالي للمتغير العشوائي x بشكل عام ما يلي :

١) $F_x(\infty)=1$ و $F_x(-\infty)=0$

٢) $F_x(x)$ هي دالة لا تنازلية (non-decreasing) في المتغير x ، بمعنى أنه إذا كان $x_1 < x_2$ ، فإن $F_x(x_1) \le F_x(x_2)$

٣) إذا كان $F_x(x_0)=0$ فإن

$F_x(x)=0$ لكل $x \le x_0$

٤) $\Pr\{X > x\} = 1 - F_x(x)$

٥) $F_x(x)$ هي دالة متصلة من اليمين بمعنى أن :

$$\lim_{t \to \infty} F_x(x+t) = F_x(x)$$

٦) $\Pr\{x_1 < X \le x_x)\} = F_x(x_2) - F_x(x_1)$

أما دالة الكثافة الاحتمالية $f_x(x)$ فلها الخصائص التالية :

١) $f_x(x) \geq 0$

٢) $F_x(x) = \int_{-\infty}^{x} f_x(x)dx$

٣) $\int_{-\infty}^{\infty} f_x(x)dx = 1$

٤) $F_x(x_2) - F_x(x_1) = \int_{x_1}^{x_2} f_x(x)dx$

$$= \Pr\{x_1 < X \leq x_2\}$$

٣- بعض التوزيعات الاحتمالية الشائعة :

٣-١ التوزيعات المتقطعة :

أ- توزيع (Binomial)

ويصف المتغير العشوائي لهذا التوزيع عدد مرات تكرار تجربة ما يكون فيها احتمالية أن ناتج معين (ويسمى احتمالية النجاح) هي (p) واحتمالية أن هذا الناتج لن يحدث (احتمالية الفشل) هي (q). فإذا أجرينا هذه التجربة عدد (n) مرة، فإن احتمالية أن الناتج سيكون هو ناتج النجاح بعدد (k) مرة هي:

$$P_n(X = k) = \binom{n}{k} p^k q^{n-k}$$

حيث $p + q = 1$

ب – توزيع (Uniform)

يتصف المتغير العشوائي لهذا التوزيع بأن جميع القيم لها نفس احتمالية الحدوث . أي أن :

$$f_x(x_i) = \Pr\{X = x_i\} = \frac{1}{n} \, , x_i = 1,2,....,n$$

ت– توزيع (Poisson)

ويمثل المتغير العشوائي X لهذا التوزيع عدد الأحداث التي تقع في فترة تمثل وحدة الزمن تحت الفرضيات التالية :

١) أن جميع أوقات وقوع الأحداث متباينة .

٢) أي فترة محدودة من الزمن لا تحتوي سوى على عدد محدود من الأحداث أو مرات الوقوع.

٣) أي فترة لا نهائية من الزمن ستحتوي على عدد لانهائي من الأحداث أو مرات الوقوع .

٤) تقع الأحداث في أزمنة غير معروفة وغير محددة مسبقاً .

٥) عدد مرات الحدوث في الفترات الزمنية غير المتداخلة يكون مستقلاً عن بعضه البعض.

ولهذا التوزيع دالة الكثافة الاحتمالية التالية :

$$\Pr\{X = k\} = \frac{l^k \; e^{-l}}{k!}, k = 0,1,2,....$$

حيث (l) هو مؤشر (Parameter) يمثل متوسط عدد مرات الحدوث خلال الفترة الواحدة.

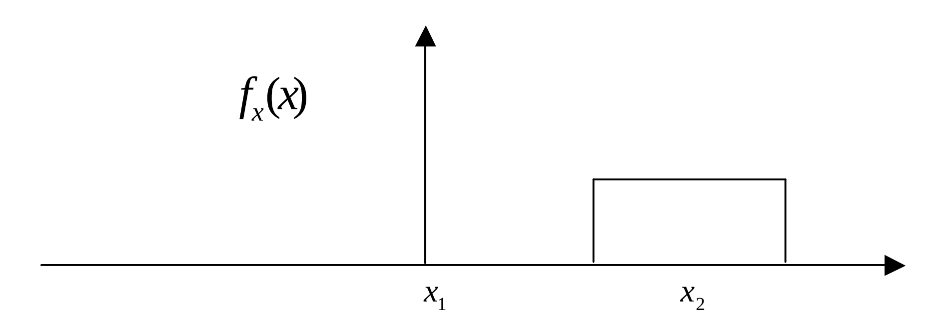
٣-٢ التوزيعات المتصلة

أ- توزيع (Uniform)

لهذا التوزيع الصيغة التالية لدالة الكثافة الاحتمالية :

$$f_x(x) = \begin{cases} \dfrac{1}{x_2 - x_1} & x_1 \le x \le x_2 \\ 0 & \text{Otherwise} \end{cases}$$

والتي يمكن تمثيلها بيانيا كالتالي :

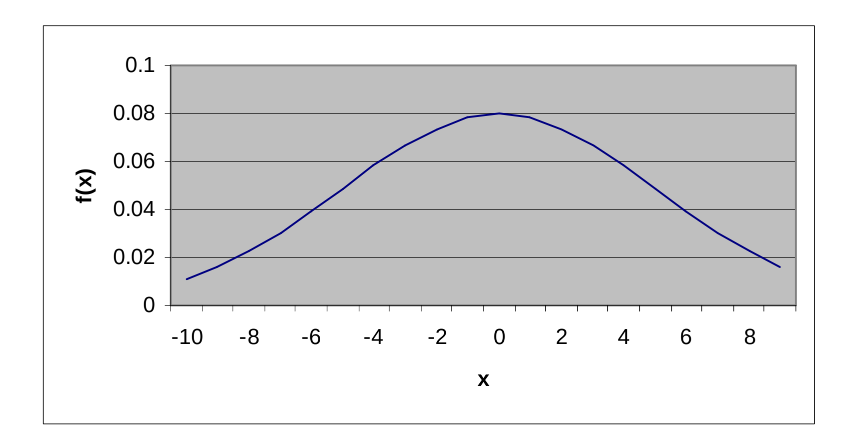


ب- توزيع (Normal)

ولهذا التوزيع الصيغة التالية لدالة الكثافة الاحتمالية :

$$f_x(x) = \frac{1}{\sqrt{2p}\,2s_x^2} e - (x - m_x)^2 / 2s_x^2$$

ويمكن تمثيل تلك الدالة بيانياً كالتالي :



حيث يمثل m_x القيمة المتوسطة للمتغير العشوائي (X) وتمثل s_x الانحراف المعياري للمتغير (X).

ت – توزيع (Exponential)

ويستخدم هذا التوزيع لتمثيل خصائص المتغيرات العشوائية التي تمثل الأزمنة البينية بين حالات الوصول للنظام أو أزمنة الخدمة لدى الخوادم في نظم الصفوف (Queuing Systems) ولهذا التوزيع الصيغة التالية لدالة الكثافة الاحتمالية :

$$f_x(x) = ae^{-ax}u(x)$$

حيث $u(x)$ هي دالة الخطوة الأحادية (Unit-Step Function) والمعرفة بالصيغة التالية:

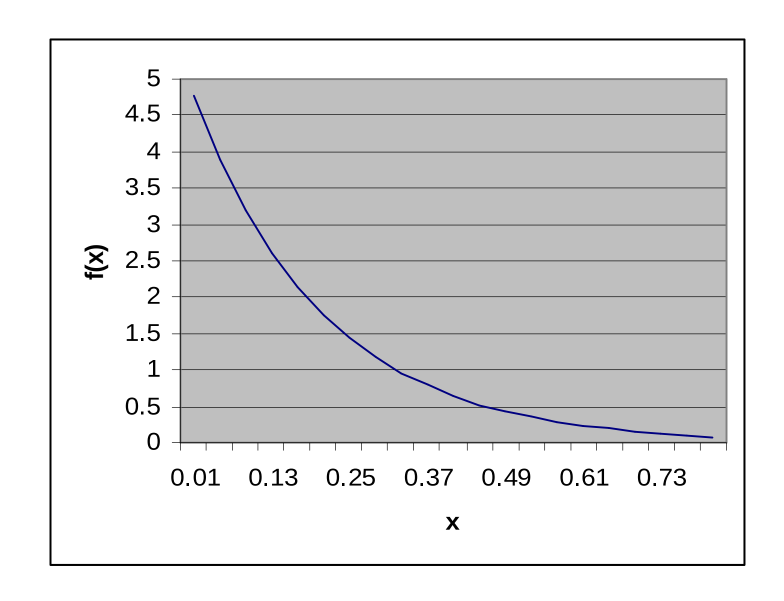
$$u\ (x) = \left[\ 0 \quad x < 0 \right.$$

$1 \quad x \geq 0$

وحيث (a) هي مقلوب قيمة المتوسط ، أي أن :

$$m_x = \frac{1}{a}$$

ولهذه الدالة الشكل التالي :



٤ – التوقع (Expectation) وحساب المتوسطات

إذا أعطينا متغير عشوائي (X) له دالة كثافة احتمالية $f_x(x)$ ، فإن القيمة المتوقعة (Expected Value) لهذا المتغير هي $E(x)$ والمعرفة كالتالي :

$$E\ (x) = \int_{-\infty}^{\infty} x f_X\ (x) dx$$

في حال ما إذا كان (X) متصلاً ، وتعرف على أنها:

$$E(x) = \sum_i x_i\, p(X = x_i)$$

في حال ما إذا كان (x) متقطعاً .

مثال :

إذا كان (X) متغيرًا عشوائيًا متصلاً وله توزيع (Uniform) بين a_1 و a_2 ، فإن $E(x)$ تساوي :

$$E(x) = \int_{-\infty}^{\infty} x f_X(x)dx = \int_{a1}^{a2} \frac{1}{a_2 - a_1} dx = \frac{a_1 + a_2}{2}$$

مثال :

إذا كان (X) متغيراً عشوائياً متقطعاً يأخذ القيم $\{1,2,3,4,5,6\}$ باحتمالية متساوية، فإنه يمكن حساب $E(x)$ كالتالي :

$$E(x) = \sum_{i=1}^{6} P_i X_i = \frac{1}{6}(1+2+3+4+5+6) = 3.5$$

ويمكن تعريف القيمة المتوقعة بشكل مشابه لأي دالة معرفة على متغير عشوائي (X) ، ولتكن $g(x)$ ، وذلك وفقاً للآتي:

$$E\{g(x)\} = \int_{-\infty}^{\infty} g(x) f_x(x)dx$$

إذا ما كان (X) متصلاً ، و

$$E\{g(x)\} = \sum_i g(x_i) P(X = x_i)$$

إذا ما كان (X) متقطعاً .

ويتسم مؤثر التوقع $E(.)$ بأنه مؤثر (Operator) خطي (Linear)، أي أن له الخاصية:

$$E\{a_1 g_1(x) + a_2 g_2(x)\}$$

$$= a_1 E\{g_1(x)\} + a_2 E\{g_2(x)\}$$

حيث إن (a_1) و a_2 هما ثابتان .

ويعرف تباين المتغير العشوائي (Variance) على أنه القيمة $Var(x)$ المعطاة وفقاً للتعريف التالي :

$$Var(X) \stackrel{\Delta}{=} E\{(X - m_x)^2\}$$

حيث $m_x \stackrel{\Delta}{=} E(X)$

ويسمى التباين أيضاً العزم الثاني أو (Second Moment) للمتغير العشوائي (X).

ويمكن إثبات أن :

$$Var(X) = E(X^2) - \{E(X)\}^2$$
$$= E(X^2) - m_x^2$$

البرهان :

$$
\begin{aligned}
Var(X) &= E(X - m_x)^2 \\
&= E(X^2 - 2m_x x + m_x^2) \\
&= E(X^2) - E(2m_x X) + E(m_x^2) \\
&= E(X^2) - 2m_x E(X) + m_x^2 \\
&= E(X^2) - 2m_x^2 + m_x = E(X^2) - m_x^2
\end{aligned}
$$

حيث أن m_x هو عدد ثابت .

مثال :

يمكن حساب قيمة (s_x) لمتغير عشوائي متصل له توزيع من نوع (Uniform) على الفترة من a_1 إلى a_2 كالآتي :

$$
\begin{aligned}
s_x^2 &= E(X^2) - m_x^2 \\
E(X^2) &= \int_{-\infty}^{\infty} x^2 f_X(x)dx = \int_{a_1}^{a_2} \frac{x^2}{a_2 - a_1} dx \\
&= \frac{a_2^3 - a_1^3}{3(a_2 - a_1)}
\end{aligned}
$$

وبما أن : $m_x = \dfrac{a_1 + a_2}{2}$

فإن :

$$
s_x^2 = \frac{a_2^3 - a_1^3}{3(a_2 - a_1)} - \left(\frac{a_1 + a_2}{2}\right)^2
$$

$$
= \left(\frac{a_2 - a_1}{12}\right)^2
$$

ويمكن تعميم تعريف العزم الثاني إلى العزم رقم r للمتغير العشوائي (X) كالتالي:

$$E(X^r) \stackrel{\Delta}{=} \int_{-\infty}^{\infty} x^r f_x(x)dx$$

إذا كان (X) متصلاً ، و

$$E(x^r) \stackrel{\Delta}{=} \sum_i x_i^r p_i$$

إذا كان (X) متقطعاً .

كما يمكن تعريف العزم المركزي رقم r (r-th Central Moment) للمتغير العشوائي (X) كما يلي:

$$E(X - m_x)^r \stackrel{\Delta}{=} \int_{-\infty}^{\infty} (x - m_x)^r f_x(x)dx$$

إذا كان (X) متصلاً ، ويعرف على أنه:

$$E(X - m_x)^r \stackrel{\Delta}{=} \sum_i (X - m_x)^r p_i$$

إذا كان (X) متقطعاً .

مثال :

أثبت المتباينة التالية (متباينة Chebyshev) :

$$\Pr\{|X - m_x| \le e\} \le \frac{s_x^2}{e^2}, e > 0$$

$$\Pr\{|X - m_x| \ge e\} \stackrel{\Delta}{=} \int_{|X - m_x| \ge e} f_x(x)dx$$

ونظراً لأن :

$$s_x^2 = E(X-m_x)^2 \overset{\Delta}{=} \int_{-\infty}^{\infty}(x-m_x)^2 f_x(x)dx$$
$$\geq \int_{|X-m_x|\geq e}(x-m_x)^2 f_x(x)dx$$
$$\geq e^2 \int_{|X-m_x|\geq e} f_x(x)dx$$

ونستنتج من ذلك ومما سبق أن :

$$\Pr\{|X-m_x| \geq e\} \leq \frac{s_x^2}{e^2}$$

٥- التوزيعات المشتركة (Joint Distributions)

يعرف التوزيع المشترك لمتغيرين عشوائيين (X) و (Y) بالدالة $F_{XY}(x,y)$ المعرفة كالتالي :

$$F_{XY}(x,y) \overset{\Delta}{=} \Pr\{X \leq x, Y \leq y\}$$

كما تعرف دالة الكثافة الاحتمالية المشتركة في حال ما إذا كانت دالة التوزيع المشترك متصلة على أنها الدالة التي تحقق :

$$\iint_Q f_{XY}(x,y) = \Pr\{X,Y) \in Q\}$$

ومن خصائص هذه الدالة ما يلي :

١) $f_{XY}(x,y) \geq 0$ لكل النقاط (x,y) في السطح R_1XR_2 . حيث إن R_1 هو نطاق المتغير العشوائي x ، و R_2 هو نطاق المتغير العشوائي Y.

٢) $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)dxdy = 1$

٣) $\int_{-\infty}^{x}\int_{\infty}^{y} f_{XY}(a,b)dadb = F_{XY}(x,y)$

٤)
$$f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y} = \frac{\partial^2 F_{XY}(x,y)}{\partial y \partial x}$$

كما أن الدالة $F_{XY}(x,y)$ لها الخصائص التالية :

١)
$$F_{XY}(-\infty,\infty) = 1$$
$$F_{XY}(-\infty,y) = 0$$
$$F_{XY}(x,-\infty) = 0$$

٢)
$$\Pr\{x_1 < X < x_2, Y \le y\} = F_{XY}(x_2,y) - F_{XY}(x_1,y)$$

٣)
$$\Pr\{x_1 < X \le x_2, y1 \le Y \le y_2\} = F_{XY}(x_2,y_2) - F_{XY}(x_1,y_2) - F_{XY}(x_2,y_1) + F_{XY}(x_1,y_1)$$

٤) $F_{XY}(x,y) = \int_{-\infty}^{x}\int_{-\infty}^{y} f(a,b)dadb$

ويمكن إيجاد علاقة بين التوزيع الاحتمالي للمتغير العشوائي (X) منفرداً، ويسمى بالتوزيع الهامشي (Marginal Distribution) وبين التوزيع المشترك كالتالي:

$$F_X(x) \overset{\Delta}{=} \Pr\{X \le x\} = \Pr\{X \le x, Y \le \infty\}$$
$$\overset{\Delta}{=} F_{XY}(x,\infty)$$

بالطريقة نفسها ، يمكن إيجاد :

$$F_Y(y) \stackrel{\Delta}{=} F_{XY}(\infty, y)$$

ملحق (ب)

تصميم تجارب المحاكاة

سنقدم فيما يلي مقدمة سريعة عن الأساليب الإحصائية المستخدمة في تصميم التجارب المعملية بصفة عامة. ونفترض في هذا الصدد أن هدف دراسة المحاكاة هو الوصول إلى أفضل مجموعة من المدخلات (Inputs) والمؤثرات (Parameters) والفرضيات الهيكلية عن النموذج التي تؤدي إلى الوصول إلى أفضل أداء للنظام المطلوب محاكاته حسب مقاييس الأداء المعتمدة. وفي هذا الصدد عادة ما يصطلح على تسمية المدخلات والمؤثرات والفرضيات الهيكلية حول النموذج بالعوامل (Factors). وبعض هذه العوامل يمكن افتراضه كثابت في النموذج لا يمكن التحكم فيه .. بينما يمكن افتراض البعض الآخر كمتغيرات قرار (Decision Variable) يمكن التحكم فيها، وذلك بحسب أهداف الدراسة. أما مؤشرات الأداء المخرجة من النظام، فعادة يصطلح لها باسم الردود (Responses)، وهي في الغالب متعددة في النموذج الواحد. ويلاحظ هنا أن العوامل في نماذج المحاكاة تنقسم إلى كمية (Quantitative) أو كيفية (Qualitative) بحسب ما إذا كانت تمثل مؤثرات ومدخلات عددية أم فرضيات هيكلية عن النظام. كما يمكن تقسيم العوامل أيضاً إلى قابلة للتحكم (Controllable) أو غير قابلة للتحكم (Uncontrollable)، وذلك بحسب ما تمثله في واقع النظام من خيارات بأيدي المديرين ومتخذي القرار.

وفي مجال المحاكاة، تعتبر عملية تصميم التجارب مرحلة مهمة من مراحل دراسة المحاكاة، إذ تقدم طريقة علمية لتحديد أفضل أوضاع النظام والتي ينبغي محاكاتها من أجل الوصول إلى الهدف المنشود من دراسة

المحاكاة بأقل عدد ممكن من مرات التشغيل لبرنامج المحاكاة. وعملية تصميم التجارب لها أساليب علمية معروفة نظراً لاستخدام التجارب والمنهجية التجريبية في كثير من العلوم سوءاً الفيزيائية أو الزراعية أو الصناعية أو غيرها. إلا أن طبيعة التجارب في مجال المحاكاة تختلف في جوانب عدة عن التجارب في المجالات الأخرى. فعلى سبيل المثال، يمكن في تجارب المحاكاة التحكم في المؤثرات التي تعتبر في النموذج "لا يمكن التحكم بها"، وذلك نظراً لأن نموذج المحاكاة نموذج حاسوبي افتراضي وليس كما هو الحال في التجارب الطبيعية. كما يمكن أيضاً التحكم في نوع وخصائص الكميات العشوائية في النظام وتغييرها حسب رغبة المستخدم عن طريق مولدات الأرقام العشوائية في برنامج المحاكاة.

وفيما يلي سوف نستعرض الأفكار الرئيسة لإحدى الطرق العلمية المستخدمة على نطاق واسع في عملية تصميم التجارب في الحالات التي يرغب الدارس فيها معرفة أهم العوامل المؤثرة في النظام والكيفية التي تؤثر بها على الردود (Responses) المختلفة للنظام. والطريقة التي وقع عليها الاختيار للاستخدام في هذا المشروع هي الطريقة المعروفة باسم (2^k Factorial) وذلك لأنها توفر دعماً كبيراً في فهم نماذج المحاكاة المعقدة وتبسيط علاقات العوامل المؤثرة فيها للدارس، مع سعة انتشارها في مجال دراسات المحاكاة (Law and Kelton, 2000).

وفي هذه الطريقة، نفترض أن نموذج المحاكاة يحتوي على عدد (K) عامل مؤثر، والمطلوب الوصول إلى تصور مبدئي عن الكيفية التي يؤثر بها كل عامل على معايير الأداء المعرفة للنظام (الردود)، وكذلك الكيفية

التـي يتفاعل بهـا العوامل مع بعضها البعض فتغير من تأثير بعضها البعض علـى ردود النظام. ولتحقيق ذلك، تقترح الطريقة تثبيت مستوى (k-1) من العوامل المؤثـرة علـى قيم محددة، وتشغيل برنامج المحاكاة لمرات عدة لكل مـن القيم المختـارة للعامـل الأول والـذي نـريد التركيـز عليه لمعرفة تأثيـره منفرداً علـى ردود النظام. بعد ذلك، يتم تكرار العملية السابقة لاختبار تأثير كل مـن العوامل الأخـرى، كل علـى حدة. ونظـراً لأن اختيار عدة قيم لكل عامل لتجـريبها عند اختبار تأثير ذلك العامل يؤدي إلى ارتفاع عدد تجارب المحاكاة (عدد مرات تشغيل البرنامج) بشكل هائل، فإن طريقة (2^k Factorial) تقترح الاكتفاء بمستويين فقط من قيم كل عامل، مما يؤدي إلى حصر عدد تجارب المحاكاة إلى العدد 2^k تجربة تمثل 2^k تركيبة (Combination) من قيم العوامل المؤثـرة فـي النظام. وعـادة مـا يُرمـز للمستوى الأدنـى لكـل عامـل بالرمـز (-) وللمستوى الأعلـى بالرمـز (+)، وغالبـاً مـا تعتمد المسافة بـين القيمتين على تقدير المصمم وخبرته في مجال عمل النظام. ويوضح الجدول رقم (١) مصفوفة التصميم للحالة عندما تكون (k = 3).

الجدول رقم (١): مصفوفة التصميم للحالة k=3

رقم المجموعة	العامل الأول	العامل الثاني	العامل الثالث	قيمة الرد R
١	-	-	-	R1
٢	+	-	-	R2
٣	-	+	-	R3
٤	+	+	-	R4
٥	-	-	+	R5
٦	+	-	+	R6
٧	-	+	+	R7

R8	+	+	+	٨

ولقياس متوسط التغيير في أحد الردود الناتجة عن تغيير عامل معين، فإننا نعرف ما يسمى بالتأثير الرئيسي (Main Effect) للعامل على أنه متوسط التغير في قيمة أحد الردود نتيجة لتغيير هذا العامل من القيمة (-) إلى القيمة (+) بينما نثبت قيم كافة العوامل الأخرى. ويعطى التأثيرات الرئيسة e1, e2, e3 لكل من العوامل الأول والثاني والثالث في حالة (k=3) المبينة في جدول (١) بالصيغة التالية (Law and Kelton, 2000) :

$$e1 = \frac{(R2 - R1) + (R4 - R3) + (R6 - R5) + (R8 - R7)}{4}$$
$$e2 = \frac{(R3 - R1) + (R4 - R2) + (R7 - R5) + (R8 - R6)}{4}$$
$$e3 = \frac{(R5 - R1) + (R6 - R2) + (R7 - R3) + (R8 - R4)}{4}$$

كما يمكن قياس درجة التفاعل بين عاملين من خلال تعريف ما يسمى بتأثير التفاعل (Interaction Effect) والذي يساوي نصف الفرق بين متوسط تأثير العامل الأول عندما يكون العامل الثاني عند قيمته (+) (وكل العوامل الأخرى ثابتة)، ومتوسط تأثير العامل الأول نفسه عندما يكون العامل الثاني عند قيمته (-). وأهمية تأثير التفاعل بين أزواج العوامل تكمن في أنها تحدد مدى إمكانية الاستفادة من قيم التأثير الرئيس للعوامل الفردية وتفسيرها حسب التعريف، حيث إن تعريف تلك العوامل ينطبق فقط على الحالات التي يكون تأثير التفاعل بينها صغيرًا نسبياً.

حالة دراسية: تصميم التجارب واستخراج النتائج في نموذج محاكاة

نظام نقل الحجاج بالحافلات .

يوضح الجدول رقم (٢) تعريفات المدخلات المستخدمة في تجارب المحاكاة لنموذج النقل بالرحلات الترددية عن طريق الحافلات في المشاعر المقدسة.

الجدول رقم (٢): مدخلات التجارب (متغيرات يمكن التحكم بها)

رمز المتغير	وصفه
ف	متوسط الزمن الفاصل بين خروج كل قافلة من الحافلات والقافلة التي تليها للمرحلة الواحدة
ع	العدد الكلي للحافلات الأساسية المشاركة فعلياً في النقل الترددي في المرحلة الواحدة
ت	مجموع متوسط أزمنة التحميل والإنزال للحجاج في المرحلة الواحدة

وسوف نعرف لهذه التجارب مقياسين للأداء هما:

١- (ن): الزمن الكلي المطلوب لإخلاء حجاج نظام الرحلات الترددية في كل مرحلة.

٢- (ز): متوسط زمن الرحلة للحافلة ذهاباً فقط في كل مرحلة.

ومقياس الأداء هو مخرج يمثل مقياساً لأداء النظام وتساعد قيمه على معرفة مدى نجاح النظام وكفاية تشغيله من قبل صناع القرار. أما المدخل فهو متغير يمكن التحكم به، أي متغير يحدد قيمته صانع القرار ويسعى لمعرفة أفضل قيم له ينتج عنها الوصول إلى أفضل أداء للنظام (أي أفضل

القيم لمقاييس الأداء). وسوف نصطلح على أن الرقم الذي يلي رمز أي من المتغيرات السابقة يشير إلى رقم المرحلة بحيث يشير الرقم ١ إلى مرحلة النقل بين عرفات ومزدلفة و رقم ٢ إلى مرحلة النقل بين مزدلفة ومنى. فعلى سبيل المثال، سوف نشير بالرمز ن١ إلى الزمن الكلي المطلوب لإخلاء كافة حجاج مرحلة عرفات-مزدلفة وتوصيلهم إلى مواقع مجموعاتهم في مزدلفة .. بينما سيشير ن٢ إلى الزمن الكلي المطلوب لإخلاء كافة حجاج مرحلة مزدلفة- منى وتوصيلهم إلى مخيماتهم في منى.

وبتطبيق منهجية (2^k Factorial) في حالة K=3 على المدخلات الثلاثة ومقياسي الأداء الواردين أعلاه، سنصل إلى جدول لكل مرحلة كما في الجدول رقم (٣) لمرحلة عرفات-مزدلفة والجدول رقم (٤) لمرحلة مزدلفة-منى. ويتم استكمال بيانات الجدولين خلال عملية استخراج البيانات من خلال حساب المتوسط الحسابي لعدد من القراءات لكل مقياس من مقاييس الأداء يمثل كل منها مرة واحدة من مرات تشغيل برنامج المحاكاة لكل حالة (سيناريو) من الحالات الناتجة عن التراكيب المختلفة من القيم المختارة من المداخل الثلاثة وعددها $٢^٣$=٨حالات. وقد تم اختيار القيمة الصغرى (-) والكبرى (+) لكل مدخل حسب القيم الموضحة في الجدولين رقمي (٣) و (٤).

الجدول رقم (٣): مرحلة عرفات-مزدلفة: جدول قيم مقاييس الأداء لبدائل 2^k Factorial في حالة K=3

البديل	متوسط زمن تقاطر القوافل ف١ (دقيقة)	عدد الحافلات المشاركة في المرحلةع ١ (حافلة)	مجموع متوسط زمني التحميل والإنزال ت١ بعرفات ومزدلفة(دقيقة)	الزمن الكلي للمرحلةن ١ (ساعة)	متوسط زمن الرحلة للحافلةز١ (دقيقة)
١	٦	٦١٨	٨,٢٧	٧,٩	٣٠,١
٢	٦	٦١٨	٤,٢٧	٧,٩	٣٦,٣
٣	٦	٤٦٦	٨,٢٧	٨,٤	٢٩,١
٤	٦	٤٦٦	٤,٢٧	٨,٤	٣٦,٩
٥	١	٦١٨	٨,٢٧	٨,٢	٣٣,٧
٦	١	٦١٨	٤,٢٧	٨	٣٥
٧	١	٤٦٦	٨,٢٧	٨,٧	٣٣
٨	١	٤٦٦	٤,٢٧	٨,٣	٣٠,٦

الجدول رقم (٤): مرحلة مزدلفة-منى: جدول قيم مقاييس الأداء لبدائل 2^k Factorial في حالة K=3

البديل	متوسط زمن تقاطر القوافل ف٢ (دقيقة)	عد الحافلات المشاركة في المرحلة ع٢ (حافلة)	مجموع متوسط زمني التحميل والإنزال ت٢ بمزدلفة	مقياس الأداء الزمن الكلي للمرحلة ن٢ (ساعة)	مقياس الأداء متوسط زمن الرحلة للحافلة ز٢

			ومنى (دقيقة)		(دقيقة)
١	٤٠،١٠	٣٣٠	٧,٨	٩,٨٣	٢٩,٨
٢	٤٠،١٠	٣٣٠	٦,٨	٩,٦٥	٢٩,٦
٣	٤٠،١٠	٢٣١	٧,٨	٩,٦٢	١٨
٤	٤٠،١٠	٢٣١	٦,٨	٩,٥٥	١٨,٥
٥	٣٠ ،٧	٣٣٠	٧,٨	٩,٦٧	٢٥,٢
٦	٣٠ ،٧	٣٣٠	٦,٨	٩,٦٩	٢٧,٦
٧	٣٠ ،٧	٢٣١	٧,٨	٩,٥٩	١٨,٤
٨	٣٠ ،٧	٢٣١	٦,٨	٩,٦٣	١٩,٨

وكما يظهر في نتائج مرحلة عرفات - مزدلفة، فإن قيم ن١ تراوحت للحالات الثماني بين ٧,٩ ساعات وبين ٨,٤ ساعات، علماً بأن قياس الزمن لمرحلة عرفات-مزدلفة يبدأ من وقت أذان المغرب من يوم عرفة. كما تراوح متوسط زمن الرحلة للحاج داخل الحافلة ز١ بين ٢٩,١ دقيقة و ٣٦,٩ دقيقة. وكما يظهر من نتائج مرحلة مزدلفة – منى في الجدول رقم (٦)، فإن قيم ن٢ تراوحت للحالات الثماني بين ٩,٥٥ و٩,٨٣ ساعات من منتصف الليل في الحالات الموصوفة في هذا الجدول. كما تراوح متوسط زمن الرحلة للحاج داخل الحافلة ز٢ بين ١٨ دقيقة و ٢٩,٨ دقيقة.

وبعد استكمال بيانات تلك الجداول من واقع مخرجات المحاكاة، يتم التعويض بقيم ن و ز من هذا الجدول في المعادلات الواردة أعلاه التي تعطي تقديرات عن التأثير الرئيس لكل عامل على حدة على كل من مقاييس الأداء (Law and Kelton, 2000). وكما هو مبين في الجدول رقم (٥) فإن التأثير الرئيس لزيادة زمن تقاطر الحافلات من ١ دقيقة إلى ٦ دقائق هو

انخفاض طفيف مقداره ٨,٦ دقائق في المتوسط لإجمالي زمن إخلاء عرفات من الحجاج، مما يدلل على عدم أهمية تأثير هذا العامل على زمن الإخلاء. أما رفع عدد الحافلات المشاركة في المرحلة من قيمة صغرى ٤٦٦ حافلة إلى قيمة عظمى ٦١٨ حافلة فيؤثر بخفض زمن إخلاء عرفات بمقدار قدره حوالي نصف ساعة في المتوسط. أما خفض مجموع متوسط زمني التحميل في عرفات والإنزال في مزدلفة للحافلة الواحدة من ٨,٢٧ دقائق إلى ٤,٢٧ دقائق فيصاحبه خفض نسبي في الزمن اللازم لإخلاء عرفات من الحجاج قدره ٨,٦ دقائق أي يقترب من ٢% من الزمن اللازم للإخلاء.

الجدول رقم (٥): مرحلة عرفات–مزدلفة: جدول التأثيرات على مقاييس الأداء

التأثير الرئيس لزيادة زمن التقاطر (ف) على زمن المرحلة (ن)	-٨,٦ دقائق
التأثير الرئيس لزيادة عدد الحافلات (ع) على زمن المرحلة (ن)	-٢٧ دقيقة
التأثير الرئيس لزيادة مجموع أزمنة التحميل والإنزال (ت) للمرحلة على زمن المرحلة (ن)	٨,٦ دقائق
التأثير الرئيس لزيادة زمن التقاطر (ف) على متوسط زمن الرحلة (ز)	٠ دقيقة
التأثير الرئيس لزيادة عدد الحافلات (ع) على متوسط زمن الرحلة (ز)	١,٤ دقيقة
التأثير الرئيس لزيادة مجموع أزمنة التحميل والإنزال (ت) للمرحلة على متوسط زمن الرحلة (ز)	-٣,٢ دقائق

وكما هو مبين أيضاً في الجدول رقم (٥) فإنه لا يوجد أي تأثير رئيس لزيادة زمن تقاطر الحافلات من ١ دقيقة إلى ٦ دقائق على متوسط زمن الرحلة للحافلة الواحدة من وقت التحرك من بوابة عرفات إلى وقت التوقف أمام بوابة مزدلفة. وتؤكد هذه النتيجة على ضآلة تأثير زيادة زمن التقاطر بين قوافل الحافلات على أداء النظام سواء زمن إخلاء عرفات من الحجاج أو متوسط زمن الرحلة من عرفات إلى مزدلفة للحافلة الواحدة. أما رفع عدد الحافلات المشاركة في المرحلة من قيمة صغرى ٤٦٦ حافلة إلى قيمة

عظمى ٦١٨ حافلة فتشير النتائج إلى أن له تأثيرًا سلبيًّا طفيفًا جداً من خلال رفع متوسط زمن الرحلة من عرفات إلى مزدلفة للحافلة الواحدة بمقدار ١,٤ دقيقة. أما خفض مجموع متوسط زمني التحميل في عرفات والإنزال في مزدلفة للحافلة الواحدة من ٨,٢٧ دقائق إلى ٤,٢٧ دقائق فيظهر من النتائج المستخرجة من المحاكاة أنه يحدث خفضاً طفيفاً في حدود ٣ دقائق في قيمة متوسط زمن الرحلة من عرفات إلى مزدلفة للحافلة الواحدة، وهو ما يمكن توقعه نظرًا؛ لأن الطريق سيخف الضغط عليه كلما طالت فترة انتظار الحافلات عند البوابات.

الجدول رقم (٦): مرحلة مزدلفة – منى: جدول التأثيرات على مقاييس الأداء

التأثير الرئيس لزيادة زمن التقاطر (ف) على زمن المرحلة (ن)	١,١ دقيقة
التأثير الرئيس لزيادة عدد الحافلات (ع) على زمن المرحلة (ن)	٦,٦ دقائق
التأثير الرئيس لزيادة مجموع أزمنة التحميل والإنزال (ت) للمرحلة على زمن المرحلة (ن)	٢,٨ دقيقة
التأثير الرئيس لزيادة زمن التقاطر (ف) على زمن الرحلة (ز)	١,٢ دقيقة
التأثير الرئيس لزيادة عدد الحافلات (ع) على زمن الرحلة (ز)	٩,٤ دقائق
التأثير الرئيس لزيادة مجموع أزمنة التحميل والإنزال (ت) للمرحلة على زمن الرحلة (ز)	-١,١ دقيقة

وكما هو مبين في الجدول رقم (٦) لمرحلة مزدلفة – منى، فإن التأثير الرئيس لزيادة زمن تقاطر الحافلات من ٣٠ دقيقة لحافلات منتصف الليل و ٧ دقائق لحافلات الساعة ٣ صباحاً إلى ٤٠ دقيقة لحافلات منتصف الليل و ١٠ دقائق لحافلات الساعة ٣ صباحاً يبلغ فقط دقيقة واحدة بالزيادة على الزمن الكلي للمرحلة، أي أنه ليس من العوامل ذات التأثير على سرعة إنهاء هذه المرحلة بإخلاء مزدلفة من الحجاج. كما يلاحظ كذلك أن زيادة عدد

الحافلات الأساسية للمرحلة من ٢٣١ إلى ٣٣٠ حافلة لا يؤدي إلى خفض ملموس في زمن إنهاء المرحلة حيث لا يبلغ سوى حوالي ٦,٦ دقائق فقط. أما عامل زمن التحميل والإنزال فهو أيضاً ليس له تأثير يذكر على زمن إنهاء المرحلة حيث يبلغ تأثيره ٢,٧ دقيقة فقط عند رفع هذا المجموع من ٦,٨ دقائق إلى ٧,٨ دقائق. أما تأثير زيادة تلك العوامل بنفس القيم المذكورة آنفاً على متوسط زمن الرحلة للحافلة الواحدة من بوابة مزدلفة إلى مخيم منى فهو على التوالي: ١,٢ دقيقة بالزيادة في حالة زيادة زمن التقاطر، ٩,٤ دقائق بالزيادة في حالة زيادة عدد الحافلات، و-١ دقيقة في حالة زيادة مجموع متوسط زمني التحميل والإنزال. وتتفق هذه النتائج مع النتائج المستخرجة لمرحلة عرفات – مزدلفة، حيث يؤثر من هذه العوامل الثلاثة عامل واحد فقط في متوسط زمن الرحلة للحافلة الواحدة وهو عامل عدد الحافلات والذي تزيد قيمة متوسطه كلما زدنا عدد الحافلات الأساسية المشاركة في النقل.

ونخلص من ذلك إلى أن زمن التقاطر في مرحلة مزدلفة – منى ليس من العوامل المؤثرة على متوسط زمن الرحلة، نظراً لوجود عوامل أخرى خارجية عن النظام أكثر تأثيراً مثل عامل قطع المشاة للمسار عرضياً. كما نستنتج أيضاً أن زيادة عدد الحافلات الأساسية المشاركة في المرحلة من القيمة (-) إلى القيمة (+) على الرغم من أنه لا يؤدي إلى زيادة تذكر في زمن إنهاء المرحلة؛ إلا أنه ينعكس سلباً بإطالة زمن الرحلة على الحجاج داخل الحافلات، حيث يساهم في زيادة زمن الرحلة في المتوسط بمقدار يبلغ ٩,٤ دقائق تقريباً. لذا، فإن عنصر عدد الحافلات المشاركة يجب أن يحدد

بعناية. وفي حالة مرحلة مزدلفة – منى، فإنه يظهر من نتائج المحاكاة أنه بالإمكان خفض هذا العدد عما هو مستخدم حالياً بدون تأثير يذكر على وقت إنهاء المرحلة بإخلاء مزدلفة من الحجاج، طمعاً في تخفيف الزحام وتقليل متوسط زمن الرحلة من مزدلفة إلى منى على الحجاج الراكبين. وأخيراً، نستنتج من النتائج السابقة أن مجموع متوسط زمني التحميل والإنزال – وكما هو متوقع – لا يكاد يكون له علاقة بمتوسط زمن الرحلة.

تعريف بالمؤلف:

د . حسام بن محمد رمضان

- أستاذ مشارك بقسم نظم المعلومات كلية علوم الحاسب والمعلومات جامعة الملك سعود.
- عمل وكيلاً لكلية علوم الحاسب والمعلومات جامعة الملك سعود منذ عام ١٤٢٢هـ.
- عمل رئيساً لقسم نظم المعلومات بالكلية من ١٤١٨هـ إلى ١٤٢٢ هـ.
- حصل على درجة الدكتوراه في علوم وهندسة الحاسب من جامعة لويفيل بولاية كنتاكي بالولايات المتحدة الأمريكية في ١٩٩٥م.
- له ما يزيد عن الثلاثين من الأبحاث العلمية المحكمة والمنشورة في مجالات: النمذجة والمحاكاة الحاسوبية وتطبيقاتها، والمحاكاة الحاسوبية المتوازية والموزعة، ونظم دعم اتخاذ القرار.
- عمل مستشاراً غير متفرغ في مجال نظم المعلومات وتقنية المحاكاة الحاسوبية للعديد من الجهات الحكومية والخاصة.
- شارك في رئاسة وعضوية فرق عمل استشارية وعلمية للعديد من الجهات الحكومية والخاصة.

عضو رفيع في كل من: جمعية المحاكاة الحاسوبية الدولية (**SCS**) وجمعية مهندسي الكهرباء والإلكترونيات الأمريكية (**IEEE**).

